CHAPTER 3

THE FINITE ELEMENT METHOD

3.1 OBJECTIVES

- Show the basic principles of the discretization of space using finite elements.
- Establish the means to obtain the integral forms of the conservation equations and to discretize them.
- Develop some aspects of the treatment of non stationary problems; non linear problems are dealt with in chapter 4.

As we have seen in chapter 1, the physical analysis of a phenomenon leads to modeling with local partial differential equations. When it is possible to employ global energy balances, these equations can be integrated over the domain in consideration, making volume contributions and boundary conditions appear.

In chapter 2, we saw that the finite difference method (FDM) was well adapted to the solution of local equations. Integrated over small volume elements, these equations can also be solved by the finite volume method (FVM). As for the finite element method (FEM), it is a powerful method to solve the equations in integral form. Two possibilities exist for application of the FEM. In the first case, there exists an integral form of the physical problem to solve: this can be a functional that results from a variational principle, the minimum of which corresponds to the desired solution, as in elasticity or visco-plasticity, or more generally, an integral equation to solve. In the second case, an integral formulation must be obtained from an initial system of partial differential equations by a weak formulation, also called the ***weighted residual method.***

With respect to the FDM or the FVM, the FEM is distinguished by its more general character, which makes it more capable of dealing with complex geometries, allows it to use non structured grids and allows more natural imposition of the boundary conditions.

Although the formalism of the FEM can appear heavier than that introduced in the previous chapter for the FDM, the principle is simple; the objective of this chapter is to show the fundamentals of the method. In the first section we introduce the basic tools for the formulation of a finite element problem: discretization of the domain, function interpolation, integration and derivation. This formalism will then be applied to a stationary scalar diffusion problem, and then to a vector problem of

mechanical equilibrium. The implementation will be discussed next, excepting the general aspects in common with the FDM which are treated in chapter 4. Finally, the method will be extended to non stationary problems.

Note that it is not our intention, in this introduction to the method, to enter into the mathematical formalism, even though it is indispensable to obtain the convergence properties of the numerical solution and the associated error estimates. The interested reader is referred to the bibliography.

3.2 GENERAL PRINCIPLES: GEOMETRIC DISCRETIZATION AND INTEGRATION

3.2.1 The discretization of space into finite elements

The principle of the FEM consists first of decomposing, or tiling, the spatial domain under consideration into a set of ***elements*** of arbitrary shape and size. This decomposition is commonly called a ***grid*** or a ***mesh***. The only restriction is that elements cannot overlap nor leave any zone of the domain uncovered. A schematic example of a two dimensional mesh is shown in figure 3.1 while figure 3.2 shows a three dimensional mesh used to solve a real problem.

As can be seen in figure 3.1, the boundary of the domain is approximated by a polygonal contour. Nonetheless, contrary to the FDM, for which the grid follows a coordinate system, the definition of a mesh for the FEM is much more free. That allows the form of the domain to be approximated with very few elements. As an aside, note that elements with non linear edges (for example, second order curves) can be used.

For each element, a certain number of points must be defined which can be situated on the edges of the element or inside. These points, called the ***nodes***, will be used to construct the approximations of the functions under consideration over the whole domain by *interpolation* (next section). As in the previous chapter, we will be

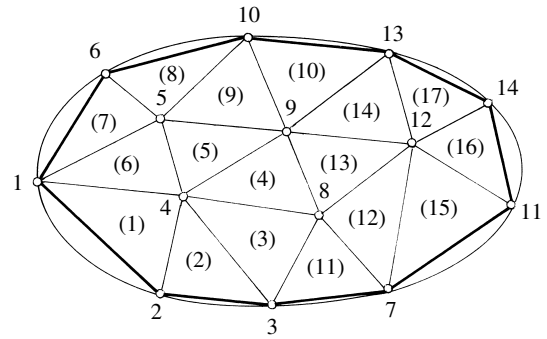


Figure 3.1 Subdivision of a geometric domain into finite elements.

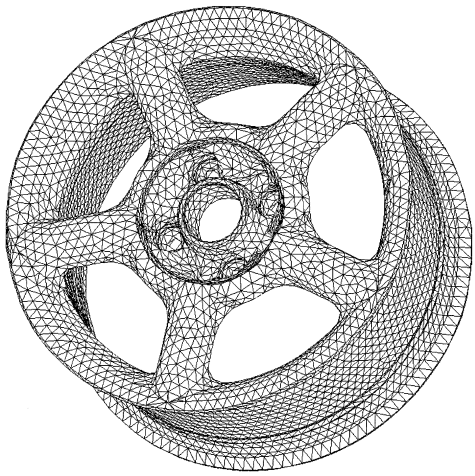


Figure 3.2 Finite element mesh of an automobile wheel rim.

led to determine the value of each unknown function or field in the problem at these points. The number of unknowns at each node is the number of nodal ***degrees of freedom***.

In the case of figure 3.1, we have assumed for simplicity that the nodes are the vertices of the triangle elements. There are then 17 elements and 14 nodes. In the following, the node index will be noted with "n" or "i", "j", "k", whereas the elements will be preferentially noted with "e".

3.2.2 The approximation of functions by interpolation in the finite elements

The approximation of a function $u(\boldsymbol{x})$ ($\boldsymbol{x}$ is a vector of spatial coordinates) on a geometric domain meshed with finite elements is obtained as a linear combination of ***interpolation functions***, $\psi_{\mathrm{n}}(\boldsymbol{x})$, associated with the mesh, thus known a priori. If $u_{\mathrm{h}}(\mathbf{x})$ is the approximation of the function $u(\boldsymbol{x})$ under consideration, it can be expressed in the form of a sum over the Nn nodes of the domain:

$$u_{\mathrm{h}}(\boldsymbol{x}) = u^{\mathrm{n}}\psi_{\mathrm{n}}(\boldsymbol{x}) \tag{3.1}$$

in which the convention is employed whereby repeated indices are summed (as throughout this chapter, except where noted). The functions $\psi_{\mathrm{n}}(\boldsymbol{x})$, used to approximate the different unknown functions in the problem, can be of diverse forms, with different degrees of continuity and differentiability. However, in the standard finite element method that we present here, the interpolation functions are defined locally at the level of each element. Thus if the node n belongs to element e, and if we use $\psi^{\mathrm{e}}_{\mathrm{n}}$ to denote the restriction of ψ_{n} within the element, we have for every coordinate

vector $\boldsymbol{x}$,

$$\psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x}) = 0 \qquad \text{if } \boldsymbol{x} \text{ is outside (strictly) the element } e, \tag{3.2}$$

and thus for $\boldsymbol{x}$ inside (including the boundary) the element e, we have:

$$u_{\mathrm{h}}(\boldsymbol{x}) = \sum_{\mathrm{n}=1}^{\mathrm{Nn}} u^{\mathrm{n}} \psi_{\mathrm{n}}(\boldsymbol{x}) = \sum_{\mathrm{n} \in \mathrm{e}} u^{\mathrm{n}} \psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x}) \tag{3.3}$$

The last sum is over the *nne* nodes that constitute the element e (for example $nne = 3$ for the case of the triangles in figure 3.1). The interpolation used for the approximation is thus *locally defined at the level of each finite element*. This elementary decomposition distinguishes the standard finite element method from other methods using interpolation functions defined over the whole of the domain; for example, spectral methods.

Also, in the standard finite element method, the coefficients u^{n} are the values of the function u_{h} at the nodes of the mesh. In consequence, the interpolation functions must respect the following two conditions, in addition to (3.2):

- If n and p are two nodes of the same element e, $\boldsymbol{x}^{\mathrm{p}}$ being the position vector of the latter node,

  $$\psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x}^{\mathrm{p}}) = \delta_{\mathrm{np}} \tag{3.4}$$

  where δ_{np} is the Kronecker delta function.
- A second condition comes from the necessity to exactly represent constant functions:

  $$\sum_{\mathrm{n} \in \mathrm{e}} \psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x}) = 1 \qquad \text{for all } \boldsymbol{x} \text{ inside the element } e \text{ (border included).} \tag{3.5}$$

3.2.3 One dimensional finite elements

We have the conditions for the partition of the domain into finite elements and the fundamental properties of the interpolation functions. To clarify these concepts, we will study in detail a one dimensional example of finite elements. This will permit the introduction of the fundamental concept of *reference elements*, or ***parent elements***. In one dimension, a finite element is necessarily reduced to a straight line segment. An example is shown in figure 3.3, where the domain $[0, L]$ is partitioned into *Ne* elements.

Linear elements

The simplest finite elements have two nodes, at the extremities of each element, and the interpolation functions are linear on these elements. Referring to

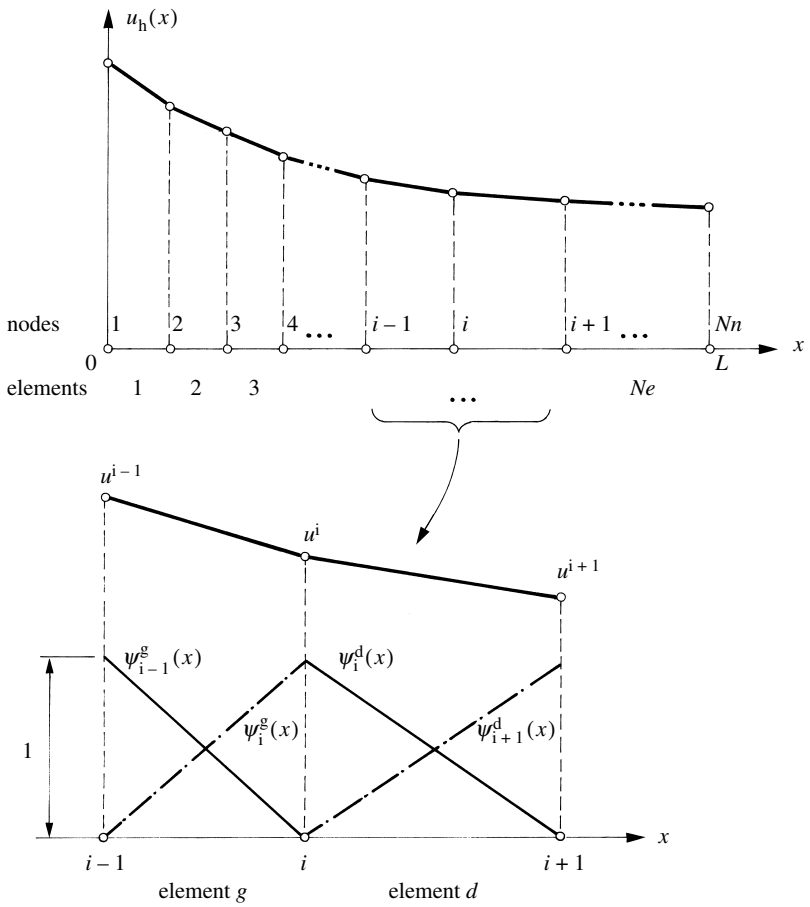


Figure 3.3 Linear elements with two nodes and interpolation functions.

figure 3.3, note element "*g* ", that which contains the nodes $i-1$ and i, and "*d*" the adjacent element with the nodes i and $i+1$. These two elements have the node i in common, and their respective lengths are Δx_g and Δx_d. Taking into account the linearity of the interpolation functions, and the conditions (3.2, 3.4 and 3.5) that they must satisfy, their expression is easily found. With the notation that x^{i-1}, x^i, x^{i+1} are the coordinates at the nodes i - 1, i and i + 1:

$$\psi^g_{i-1}(x) = \frac{x^i - x}{\Delta x_g} \text{ and } \psi^g_i(x) = \frac{x - x^{i-1}}{\Delta x_g} \text{ in the element } g: [x^{i-1}, x^i] \quad (3.6)$$

$$\psi^d_i(x) = \frac{x^{i+1} - x}{\Delta x_d} \text{ and } \psi^d_{i+1}(x) = \frac{x - x^i}{\Delta x_d} \text{ in the element } d: [x^i, x^{i+1}] \quad (3.7)$$

The global interpolation function, ψ_i, associated with node i and defined on the entire grid, is obtained by assembling ψ_i^g and ψ_i^d, as illustrated in figure 3.3. It is zero for $x \geq x^{i+1}$ or $x \leq x^{i-1}$, that is, in every element that does not contain the node i. Because of the form it thus takes, the function ψ_i is often called a 'hat function'.

Defining a local coordinate ξ in the interior of the element g, with the relation

$$\xi = \frac{x - x^{i-1}}{\Delta x_g} \tag{3.8}$$

the interpolation functions on this element take the form

$$\psi_{i-1}^g = 1 - \xi \quad \text{et} \quad \psi_i^g = \xi \tag{3.9}$$

This can of course be generalized to the element d and to every element $[x^i, x^{i+1}]$, by defining

$$\xi = \frac{x - x^i}{x^{i+1} - x^i} \tag{3.10}$$

Thus, for every element e, a mapping E_e can be defined between the element in physical space and the segment [0, 1] (fig. 3.4). This mapping allows us to define the interpolation functions universally for the diverse elements regardless of their

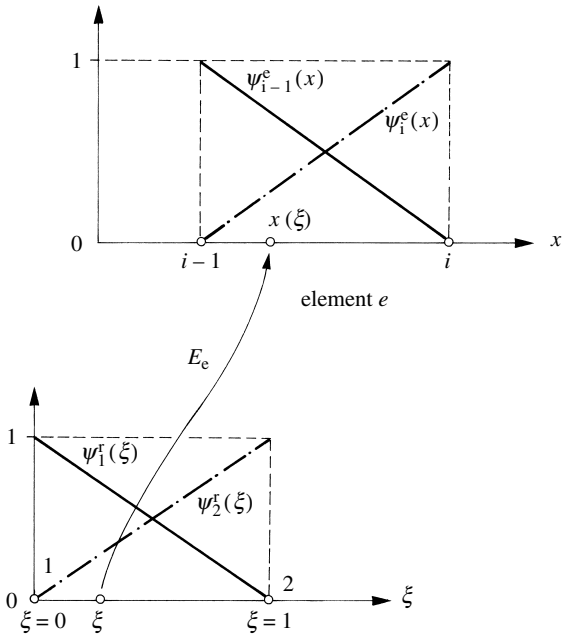


Figure 3.4 The notion of the reference element.

coordinates. The interpolation functions depend only on the nature of the element and the number of points that they contain. This is the fundamental notion of the ***reference element***.

From this reference element composed of nodes 1, at coordinate $\xi = 0$, and 2, at coordinate $\xi = 1$, the interpolation functions in physical space are easily reconstructed; denoting ν_k^e the global numbering of node k (local) in the element e ($k = 1, 2$), we have for the preceding example:

$$\nu_1^g = i - 1 \qquad \text{and} \quad \nu_2^g = i \qquad \text{in the element } g \tag{3.11}$$

$$\nu_1^d = i \qquad \text{and} \quad \nu_2^d = i + 1 \qquad \text{in the element } d \tag{3.12}$$

Denoting ψ_1^r and ψ_2^r the interpolation functions associated with nodes 1 and 2 of the reference element, and ξ being given by (3.10), we have:

$$\psi_{i-1}^g(x) = \psi_1^r(\xi(x)) = 1 - \xi(x) \text{ and } \psi_i^g(x) = \psi_2^r(\xi(x)) = \xi(x) \tag{3.13}$$
(element g)

$$\psi_i^d(x) = \psi_1^r(\xi(x)) = 1 - \xi(x) \text{ and } \psi_{i+1}^d(x) = \psi_2^r(\xi(x)) = \xi(x) \tag{3.14}$$
(element d)

The function $u(x)$ is approximated in element g by the following linear combination:

$$u_h(x) = u^{i-1}\psi_{i-1}^g(x) + u^i\psi_i^g(x) = u^{i-1} + \frac{x - x^{i-1}}{\Delta x_g}(u^i - u^{i-1}) \tag{3.15}$$

for $x \in [x^{i-1}, x^i]$.

Similarly, in element d, we have:

$$u_h(x) = u^i\psi_i^d(x) + u^{i+1}\psi_{i+1}^d(x) = u^i + \frac{x - x^i}{\Delta x_d}(u^{i+1} - u^i) \tag{3.16}$$

for $x \in [x^i, x^{i+1}]$.

Using (3.11), the mapping relation between the local (element level) node numbering and the global numbering, we can write for each element e:

$$u_h(x) = u^n\psi_n^e(x) = u^{\nu_k^e}\psi_k^r(\xi(x)) \tag{3.17}$$

the two sums are over *nne* nodes, the number of nodes of the reference finite element, which is 2 here (n global numbering, k local numbering).

The derivative of u_h in the interior of the element e is given in this representation by the following:

$$\frac{\mathrm{d}u_\mathrm{h}}{\mathrm{d}x} = u^\mathrm{n}\frac{\mathrm{d}\psi^\mathrm{e}_\mathrm{n}}{\mathrm{d}x} = u^{\nu^\mathrm{e}_\mathrm{k}}\frac{\mathrm{d}\psi^\mathrm{r}_\mathrm{k}}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x} \tag{3.18}$$

where $\mathrm{d}\xi/\mathrm{d}x$ comes from (3.10). It can be seen that the derivatives of u_h are not necessarily continuous at the nodes. Thus to express the derivative at a node, a smoothing procedure will be required for the derivatives calculated in each of the elements to which it belongs.

Note here that because every linear function is exactly represented on the elements, it is true in particular for the mappings $\xi(x)$ and $x(\xi)$. Thus, (3.8) and (3.10) can be written as linear combinations of the interpolation functions:

$$x = x^\mathrm{n}\psi^\mathrm{e}_\mathrm{n}(x) = x^{\nu^\mathrm{e}_\mathrm{k}}\psi^\mathrm{r}_\mathrm{k}(\xi(x)) \tag{3.19}$$

In general, the elements for which the *form functions* (those that describe the geometry of the element according to (3.19)) and the interpolation functions (those that are used to approximate a function according to (3.1)) coincide are called ***isoparametric elements***.

Quadratic elements

In the case of a one dimensional element with three nodes, two at the extremities and one in the center, the interpolation functions are of second order which permits the exact representation of quadratic functions. The reference element is classically defined on the segment $[-1, +1]$ with the central node situated at $\xi = 0$, as shown in figure 3.5. The quadratic interpolation functions are defined on this reference element by:

$$\psi^\mathrm{r}_1 = -\frac{\xi}{2}(1-\xi) \qquad \psi^\mathrm{r}_2 = 1-\xi^2 \qquad \psi^\mathrm{r}_3 = \frac{\xi}{2}(1+\xi) \tag{3.20}$$

with $\xi = 2(x - x^\mathrm{i})/(x^{\mathrm{i}+1} - x^{\mathrm{i}-1})$ on the element $[x^{\mathrm{i}-1}, x^{\mathrm{i}+1}]$.

It can be verified that these quadratic functions satisfy the conditions (3.4) and (3.5). For isoparametric elements, the mapping between x and ξ is of second order and is defined, for x in $[x^{\mathrm{i}-1}, x^{\mathrm{i}+1}]$, by the interpolation functions:

$$\begin{aligned} x(\xi) &= x^{\mathrm{i}-1}\psi^\mathrm{e}_{\mathrm{i}-1}(x) + x^\mathrm{i}\psi^\mathrm{e}_\mathrm{i}(x) + x^{\mathrm{i}+1}\psi^\mathrm{e}_{\mathrm{i}+1}(x) \\ &= x^{\mathrm{i}-1}\psi^\mathrm{r}_1(\xi) + x^\mathrm{i}\psi^\mathrm{r}_2(\xi) + x^{\mathrm{i}+1}\psi^\mathrm{r}_3(\xi) \end{aligned} \tag{3.21}$$

Similarly as for the linear elements, the derivative of a function $u_\mathrm{h}(x)$ is obtained from the relation:

$$\frac{\mathrm{d}u_\mathrm{h}}{\mathrm{d}x} = u^n\frac{\mathrm{d}\psi^\mathrm{e}_\mathrm{n}}{\mathrm{d}x} \tag{3.22}$$

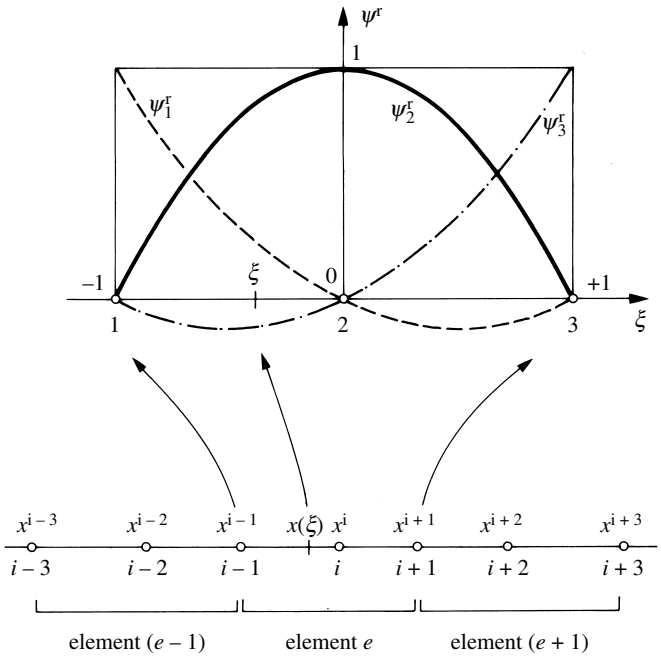


Figure 3.5 Quadratic elements.

in which the derivatives of the interpolation functions are calculated as follows (the node n corresponding in the reference element to the node k; $n = v_k^e$):

$$\frac{d\psi_n^e}{dx} = \frac{d\psi_k^r}{dx} = \frac{d\psi_k^r}{d\xi}\frac{d\xi(x)}{dx} = \frac{d\psi_k^r}{d\xi}\frac{1}{x^{v_m^e}\, d\psi_m^r/d\xi} \tag{3.23}$$

In the example in figure 3.5, we have:

$$\begin{aligned}\frac{dx}{d\xi} &= x^{i-1}\frac{d\psi_1^r}{d\xi} + x^i\frac{d\psi_2^r}{d\xi} + x^{i+1}\frac{d\psi_3^r}{d\xi} \\ &= x^{i-1}(-\tfrac{1}{2}+\xi) + x^i(-2\xi) + x^{i+1}(\tfrac{1}{2}+\xi) \\ &= \tfrac{1}{2}(x^{i+1}-x^{i-1}) - \xi((x^i-x^{i-1})-(x^{i+1}-x^i))\end{aligned} \tag{3.24}$$

For a regular grid, the right hand side reduces to $\Delta x/2$. In general, the formula (3.23) with a change of variables $x(\xi)$, permits the calculation of all the derivatives of the interpolation functions at all points of the real elements.

3.2.4 Two dimensional finite elements

As a prolongation of the introduction with one dimensional elements, a two dimensional example will further illustrate the concept of reference elements. Figure 3.6 shows the geometric transformation E_e that associates the reference element with any real element (for the triangular case):

$$E_e = \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \rightarrow \boldsymbol{x}(\boldsymbol{\xi}) = \begin{bmatrix} x_1(\boldsymbol{\xi}) \\ x_2(\boldsymbol{\xi}) \end{bmatrix} \tag{3.25}$$

This time $\boldsymbol{x}$ and $\boldsymbol{\xi}$ are vectors with two components, for which we will sometimes employ the notation (x, y) and (ξ, η). Conforming to that which was said before, E_e must satisfy the three following properties:

- Property 1: bidirectional one-to-one mapping.
- Property 2: the correspondence between the geometric node of the real element and the node of the reference element, which is written, denoting by n the global numbering of the kth local node of the element e ($n = \nu_k^e$), by:

$$\boldsymbol{x}(\boldsymbol{\xi}^k) = \boldsymbol{x}^{\nu_k^e} = \boldsymbol{x}^n \tag{3.26}$$

- Property 3: local definition of the boundaries of the real and reference elements; that is, by interpolation with only the nodes actually on the boundary in order to have a uniform representation of the boundary for an element and its neighbor.

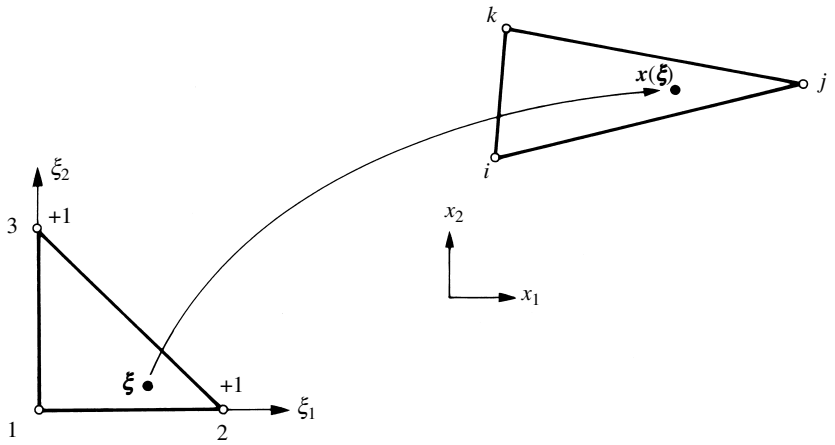


Figure 3.6 Reference element in two dimensions (linear triangle).

Given the fundamental relation

$$\boldsymbol{x}(\boldsymbol{\xi}) = \psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x})\boldsymbol{x}^{\mathrm{n}} = \psi_{\mathrm{k}}^{\mathrm{r}}(\boldsymbol{\xi})\boldsymbol{x}^{\nu_{\mathrm{k}}^{\mathrm{e}}} \tag{3.27}$$

the second property implies an essential property of nodal interpolation functions:

$$\psi_{\mathrm{k}}^{\mathrm{r}}(\boldsymbol{\xi}^{\mathrm{p}}) = \delta_{\mathrm{kp}} \tag{3.28}$$

In other words, the interpolation function associated with node k takes the value 1 at the node, and the value 0 at all other nodes of the reference element. Projected onto the real grid, we see that the interpolation function of a given node takes the value 1 at this node and 0 at all the other nodes of the grid (fig. 3.7).

The approximation of a function u is thus realized in each element e by:

$$u_{\mathrm{h}}(\boldsymbol{x}) = u^{\mathrm{n}}\psi_{\mathrm{n}}^{\mathrm{e}}(\boldsymbol{x}) = u^{\nu_{\mathrm{k}}^{\mathrm{e}}}\psi_{\mathrm{k}}^{\mathrm{r}}(\boldsymbol{\xi}(\boldsymbol{x})) \tag{3.29}$$

In the following we consider, except where specified, isoparametric elements. We then have also:

$$\boldsymbol{x} = \psi_{\mathrm{k}}^{\mathrm{r}}(\boldsymbol{\xi}(\boldsymbol{x}))\boldsymbol{x}^{\nu_{\mathrm{k}}^{\mathrm{e}}} \tag{3.30}$$

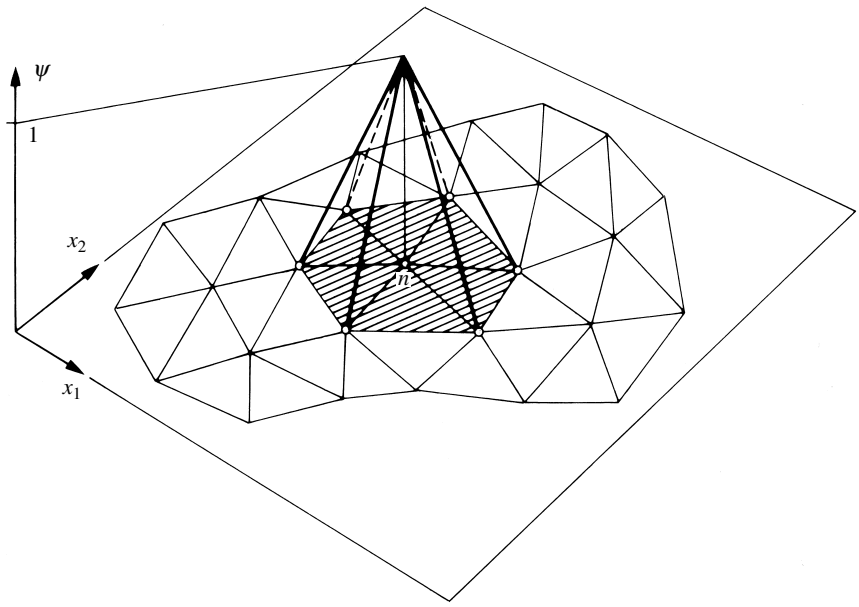


Figure 3.7 Interpolation function associated with the node n (global numbering). In this case, for linear triangular elements, the function decreases linearly from the value 1 at node n to the value 0 at all the nodes neighboring n. It is then 0 over all the elements that do not contain the node n.

Thus the interpolation functions ψ_n only depend on the reference element; and its utility becomes apparent. The approximation u_h is expressed by its nodal values and the interpolation functions of the reference element corresponding to the real element.

Note that it is not always evident how to calculate the inverse transformation $\boldsymbol{\xi}(\boldsymbol{x})$ to use (3.30); in particular, if the interpolation functions are of higher degree than second order. Nonetheless, as we will see later, the calculation of $u(\boldsymbol{x})$, and thus of $\psi_k^r(\boldsymbol{\xi}(\boldsymbol{x}))$, is only necessary at certain points in the elements: those for which the coordinates $\boldsymbol{\xi}$ are known in advance.

REMARK: The interpolation functions need not be identical for the different functions that one wants to approximate in a single problem. Such is the case, for example, of formulations in so called primitive variables, for which one solves in velocity and pressure fields $(\boldsymbol{v}, p)$. These formulations are very often used in incompressible fluid mechanics (chap. 7) and increasingly more often in solid mechanics (chap. 6).

The nodes and interpolation functions used for the approximation of the velocity components and the pressure are generally different, and one writes for example:

$$(v_h)_i(\boldsymbol{x}) = \sum_{k=1}^{nnv} \psi_k^{rv}(\boldsymbol{\xi}(\boldsymbol{x}))v_i^{v_k^e} \tag{3.31}$$

for each component i of the velocity field, nnv being the number of nodes and ψ^{rv} the functions used for interpolation of the velocity in the element. For the scalar pressure field, one has a different number of nodes and interpolation functions for the element:

$$p_h(\boldsymbol{x}) = \sum_{k=1}^{nnp} \psi_k^{rp}(\boldsymbol{\xi}(\boldsymbol{x}))p^{v_k^e} \tag{3.32}$$

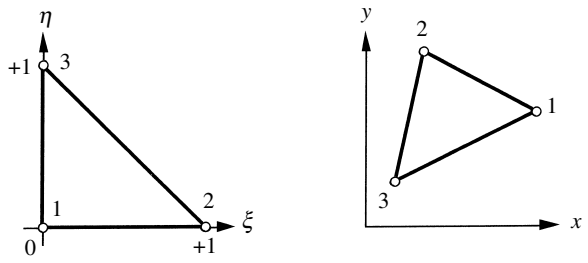
3.2.5 Examples of two dimensional finite elements

Each type of reference element is defined by:

- its form (triangular, quadrangular, …),
- the coordinates of its nodes,
- its interpolation functions,
- the definition of its nodal variables,
- the type of continuity satisfied by the approximation of a function at the element boundary (continuous function C^0, function with a continuous first derivative C^1, …).

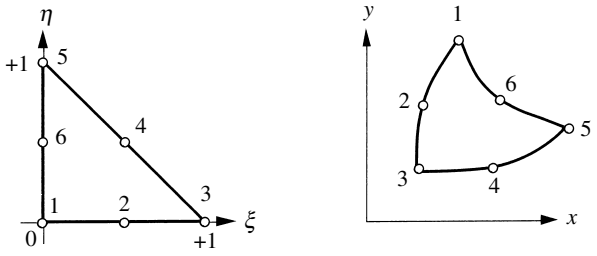
We will show a few of the most classic isoparametric finite elements. The reader is referred to the bibliography for other elements, notably those with higher continuity.

To simplify the figures, in this paragraph we indicate by (ξ, η) the coordinates in the reference element, rather than (ξ_1, ξ_2), and by (x, y) the spatial coordinates, rather than (x_1, x_2).



Node k	ψ_k
1	$1 - \xi - \eta$
2	ξ
3	η

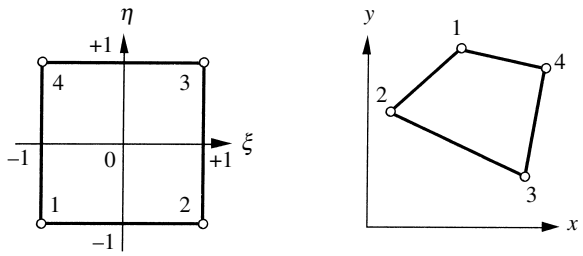
Figure 3.8 Linear triangular element (element $P1$ according to the taxonomy proposed by Ciarlet): a triangle with three nodes.



Node k	ψ_k
1	$\lambda\,(2\lambda - 1)$
2	$4\xi\lambda$
3	$\xi\,(2\xi - 1)$
4	$4\xi\eta$
5	$\eta\,(2\eta - 1)$
6	$4\eta\lambda$

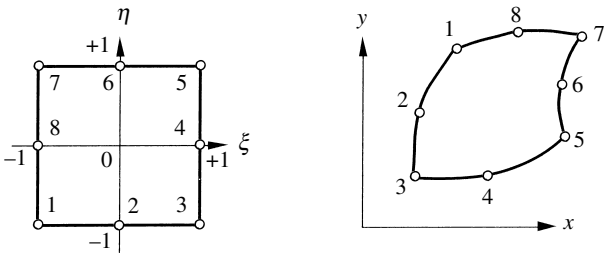
$(\lambda = 1 - \xi - \eta)$

Figure 3.9 Quadratic triangular element ($P2$): a triangle with six nodes.



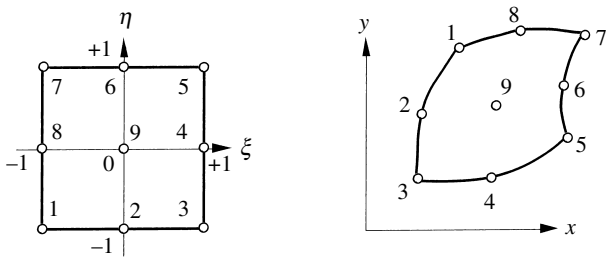
Node k	ψ_k
1	$(1-\xi)(1-\eta)/4$
2	$(1+\xi)(1-\eta)/4$
3	$(1+\xi)(1+\eta)/4$
4	$(1-\xi)(1+\eta)/4$

Figure 3.10 Linear quadrangular element ($Q1$): a quadrangle with four nodes.



Node k	ψ_k
1	$-(1-\xi)(1-\eta)(1+\xi+\eta)/4$
2	$(1-\xi^2)(1-\eta)/2$
3	$-(1+\xi)(1-\eta)(1-\xi+\eta)/4$
4	$(1+\xi)(1-\eta^2)/2$
5	$-(1+\xi)(1+\eta)(1-\xi-\eta)/4$
6	$(1-\xi^2)(1+\eta)/2$
7	$-(1-\xi)(1+\eta)(1+\xi-\eta)/4$
8	$(1-\xi)(1-\eta^2)/2$

Figure 3.11 Quadratic quadrangular element ($Q2$ incomplete): a quadrangle with eight nodes.



Node k	ψ_k
1	$\xi\eta(1-\xi)(1-\eta)/4$
2	$-\eta(1+\xi)(1-\xi)(1-\eta)/2$
3	$-\xi\eta(1+\xi)(1-\eta)/4$
4	$\xi(1+\xi)(1+\eta)(1-\eta)/2$
5	$\xi\eta(1+\xi)(1+\eta)/4$
6	$\eta(1+\xi)(1-\xi)(1+\eta)/2$
7	$-\xi\eta(1-\xi)(1+\eta)/4$
8	$-\xi(1-\xi)(1+\eta)(1-\eta)/2$
9	$(1+\xi)(1-\xi)(1+\eta)(1-\eta)$

Figure 3.12 Complete quadratic quadrangular element ($Q2$): a quadrangle with nine nodes.

3.2.6 Integration of a function

The integration of a function u on a domain Ω is first decomposed into a sum of integrals over each of the elements, each occupying a domain Ω^e, the union of which is Ω.

$$I = \int_{\Omega} u(\boldsymbol{x})\ \mathrm{d}V \approx \sum_{e=1}^{Ne} \left(\int_{\Omega^e} u_h(\boldsymbol{x})\ \mathrm{d}V \right) \tag{3.33}$$

In each element e, the change of variables $\boldsymbol{x}(\boldsymbol{\xi})$ allows the integration on the reference element, denoted Ω^r, using the determinant of the ***Jacobian matrix*** $[\boldsymbol{J}^e]$ for the change of variables. If d is the dimension of the problem (d is 2 in two dimensions, 3 in 3 dimensions), the matrix $[\boldsymbol{J}^e]$, of size $d \times d$, has the following components:

$$J^e_{ij} = \frac{\partial x_i}{\partial \xi_j} = \frac{\partial \psi^r_k}{\partial \xi_j} x_i^{v^e_k} \tag{3.34}$$

The sum k is over the *nne* nodes of the element. We have then for each element of the domain (here assumed to be two dimensional):

$$\int_{\Omega^e} u_h(\boldsymbol{x})\ \mathrm{d}V\ =\ \int_{\Omega^e} u_h(\boldsymbol{x})\mathrm{d}x_1\,\mathrm{d}x_2\ =\ \int_{\Omega^r} u_h(\boldsymbol{x}(\boldsymbol{\xi}))\ \det[\boldsymbol{J}^e(\boldsymbol{\xi})]\ \mathrm{d}\xi_1\,\mathrm{d}\xi_2 \qquad (3.35)$$

The exact calculation of the integral rapidly becomes impossible for complex forms or nonlinear elements (even if the symbolic mathematical programs can be helpful in some cases); one must turn to numerical integration techniques. The last integral can be calculated by diverse methods; the most common is ***Gauss' method***, which consists of evaluating the function u_h at judiciously chosen points in the reference element, according to the formula:

$$\int_{\Omega^r} u_h(\boldsymbol{x}(\boldsymbol{\xi}))\ \det[\boldsymbol{J}^e(\boldsymbol{\xi})]\ \mathrm{d}\xi_1\,\mathrm{d}\xi_2 \simeq \sum_{p=1}^{npie} \underbrace{u_h(\boldsymbol{x}(\boldsymbol{\xi}^p))}_{\psi_k^r(\boldsymbol{\xi}^p)u^{\nu_k^e}}\ \det[\boldsymbol{J}^e(\boldsymbol{\xi}^p)]w^p \qquad (3.36)$$

where *npie* is the number of ***integration points*** (or ***Gauss points***), with coordinate vectors $\boldsymbol{\xi}^p$ and associated weights w^p. The details of the choice of the number of points and their associated weights, which are functions of the degree of precision desired in the integral, and the formulas most often used for two dimensional finite elements are given in chapter 10. Now it is sufficient to know that, in one dimension, Gauss' method allows the exact integration of a polynomial of degree $2r-1$ with r points. For example, with two points on the segment $[-1, +1]$, having abscissas $-1/\sqrt{3}$ and $+1/\sqrt{3}$ and equal weights $(=1)$, polynomials of order 3 can be exactly integrated (chap. 10).

3.2.7 Derivatives

In the interior of element e, the derivative of the function u_h with respect to one of the coordinates x_j is written, using (3.29):

$$\frac{\partial u_h}{\partial x_j} = \frac{\partial \psi_n^e(\boldsymbol{x})}{\partial x_j} u^n \qquad (3.37)$$

or again using the transformation $\boldsymbol{x}(\boldsymbol{\xi})$,

$$\frac{\partial u_h}{\partial x_j} = \frac{\partial \psi_k^r(\boldsymbol{\xi})}{\partial x_j} u^{\nu_k^e} = \frac{\partial \psi_k^r}{\partial \xi_i}\frac{\partial \xi_i}{\partial x_j} u^{\nu_k^e} \qquad (3.38)$$

the sum over i being from 1 to d (spatial components). Denoting by $[\boldsymbol{J}^e]^{-1}$ the inverse matrix of $[\boldsymbol{J}^e]$ (which does exist; the transformation E_e is bi-directional), we have finally:

$$\frac{\partial u_h}{\partial x_j} = \frac{\partial \psi_k^r}{\partial \xi_i}(J^e)^{-1}_{ij} u^{v_k^e} \tag{3.39}$$

in which we have somewhat abusively denoted $(J^e)^{-1}_{ij}$ the component ij of the matrix $[\boldsymbol{J}^e]^{-1}$, to simplify the expression. In practice, for each element e, we calculate the matrix $[\boldsymbol{J}^e]$ at each integration point using (3.34), and take the inverse if necessary. The values of the interpolation functions and their derivatives at the integration points of the reference element are calculated once for the whole computation at the beginning.

3.2.8 Example of integration and derivation on a linear triangle

To illustrate the preceding, take as an example the calculation of an integral of the derivative of a scalar function u (the subscript "h" is omitted here), over the linear triangular element e of figure 3.8. The integral to calculate is the following:

$$I = \int_{\Omega^e} \frac{\partial u}{\partial x_1}\, \mathrm{d}x_1\, \mathrm{d}x_2 \tag{3.40}$$

The linear interpolation functions are expressed in the following manner on the reference element Ω^r (fig. 3.8):

$$\psi_1^r(\boldsymbol{\xi}) = 1 - \xi_1 - \xi_2 \qquad \psi_2^r(\boldsymbol{\xi}) = \xi_1 \qquad \psi_3^r(\boldsymbol{\xi}) = \xi_2 \tag{3.41}$$

The transformation E_e is thus:

$$\begin{aligned} E_e\colon \boldsymbol{\xi} = \begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} \to \boldsymbol{x}(\boldsymbol{\xi}) &= \psi_1^r(\boldsymbol{\xi})\boldsymbol{x}^1 + \psi_2^r(\boldsymbol{\xi})\boldsymbol{x}^2 + \psi_3^r(\boldsymbol{\xi})\boldsymbol{x}^3 \\ &= (1-\xi_1-\xi_2)\boldsymbol{x}^1 + \xi_1\boldsymbol{x}^2 + \xi_2\boldsymbol{x}^3 \end{aligned} \tag{3.42}$$

and, written as components:

$$\left.\begin{aligned} \begin{bmatrix}x_1\\ x_2\end{bmatrix} &= \psi_1^r(\boldsymbol{\xi})\begin{bmatrix}x_1^1\\ x_2^1\end{bmatrix} + \psi_2^r(\boldsymbol{\xi})\begin{bmatrix}x_1^2\\ x_2^2\end{bmatrix} + \psi_3^r(\boldsymbol{\xi})\begin{bmatrix}x_1^3\\ x_2^3\end{bmatrix} \\ &= (1-\xi_1-\xi_2)\begin{bmatrix}x_1^1\\ x_2^1\end{bmatrix} + \xi_1\begin{bmatrix}x_1^2\\ x_2^2\end{bmatrix} + \xi_2\begin{bmatrix}x_1^3\\ x_2^3\end{bmatrix} \\ &= \begin{bmatrix}x_1^1\\ x_2^1\end{bmatrix} + \begin{bmatrix}x_1^2 - x_1^1 & x_1^3 - x_1^1\\ x_2^2 - x_2^1 & x_2^3 - x_2^1\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} \end{aligned}\right\} \tag{3.43}$$

The Jacobian matrix of the transformation E_e is given by:

$$[\boldsymbol{J}^e] = \left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right] = \begin{bmatrix} \partial x_1/\partial \xi_1 & \partial x_1/\partial \xi_2 \\ \partial x_2/\partial \xi_1 & \partial x_2/\partial \xi_2 \end{bmatrix} = \begin{bmatrix} x_1^2 - x_1^1 & x_1^3 - x_1^1 \\ x_2^2 - x_2^1 & x_2^3 - x_2^1 \end{bmatrix} \tag{3.44}$$

and its determinant by:

$$\det[\boldsymbol{J}^e] = (x_1^2 - x_1^1)(x_2^3 - x_2^1) - (x_2^2 - x_2^1)(x_1^3 - x_1^1) \tag{3.45}$$

It is seen that the Jacobian matrix is constant on the element, and that its determinant is, except for a factor of 2, the area of the triangle. This is of course a particularity of linear triangular elements. In general, the matrix $[\boldsymbol{J}^e]$ varies over the element and it should be rigorously noted by $[\boldsymbol{J}^e(\boldsymbol{\xi})]$. In the present case we denote it with $[\boldsymbol{J}^e]$, and then the inverse is $[\boldsymbol{J}^e]^{-1}$:

$$[\boldsymbol{J}^e]^{-1} = \begin{bmatrix} \dfrac{\partial \xi_1}{\partial x_1} & \dfrac{\partial \xi_1}{\partial x_2} \\ \dfrac{\partial \xi_2}{\partial x_1} & \dfrac{\partial \xi_2}{\partial x_2} \end{bmatrix} \tag{3.46}$$

which contains in the first column the necessary derivatives for the expression of the derivative of u that interests us. We have:

$$\frac{\partial u}{\partial x_1} = \frac{\partial u}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_1} + \frac{\partial u}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_1} = (J^e)^{-1}_{11}\frac{\partial u}{\partial \xi_1} + (J^e)^{-1}_{21}\frac{\partial u}{\partial \xi_2} \tag{3.47}$$

Moving on to the calculation of the integral I over the reference element, by the change of variables E_e:

$$I = \int_{\Omega^e} \frac{\partial u}{\partial x_1}\, \mathrm{d}x_1\, \mathrm{d}x_2 = \int_{\Omega^r} \left((J^e)^{-1}_{11}\frac{\partial u}{\partial \xi_1} + (J^e)^{-1}_{21}\frac{\partial u}{\partial \xi_2} \right) \det[\boldsymbol{J}^e]\, \mathrm{d}\xi_1\, \mathrm{d}\xi_2 \tag{3.48}$$

in which the expression of the derivatives of u are the following (assuming for simplicity here that the numbering coincides between $\boldsymbol{\Omega}^e$ and $\boldsymbol{\Omega}^r$):

$$\frac{\partial u}{\partial \xi_1} = \sum_{n=1}^{3} \frac{\partial \psi_n^r}{\partial \xi_1} u^n \qquad \frac{\partial u}{\partial \xi_2} = \sum_{n=1}^{3} \frac{\partial \psi_n^r}{\partial \xi_2} u^n \tag{3.49}$$

Numerical integration by Gauss' method gives:

$$I = \sum_{p=1}^{npie} w^p \det[\boldsymbol{J}^e] \left[\sum_{n=1}^{3} u^n \left((J^e)^{-1}_{11} \frac{\partial \psi^r_n(\boldsymbol{\xi}^p)}{\partial \xi_1} + (J^e)^{-1}_{21} \frac{\partial \psi^r_n(\boldsymbol{\xi}^p)}{\partial \xi_2} \right) \right] \tag{3.50}$$

It should be noted here that the function u interpolated over the finite element is linear, and thus that the derivatives are constant. As a result, the integration only requires a single point in the triangle (chap. 10). The reader can verify it with a quick calculation whose result is (exercise 3.7.1):

$$I = \tfrac{1}{2}[(x^3_2 - x^1_2)(u^2 - u^1) + (x^1_2 - x^2_2)(u^3 - u^1)] \tag{3.51}$$

In conclusion, such combinations of calculations are very frequently encountered in finite element programs, and require the calculations of the following quantities in each element for each integration point $\boldsymbol{\xi}^p$:

- The matrix $[\boldsymbol{J}^e(\boldsymbol{\xi}^p)]$, its determinant and its inverse $[\boldsymbol{J}^e]^{-1}$. These values depend on the real coordinates; in the general case they must be calculated for each real element e. If the coordinates of the real element change during the problem (as in a study of solid deformation, for example), it will be necessary to recalculate them every time they change.
- The functions ψ_n and their derivatives in the reference space. Nonetheless, these values only depend on the reference element; they need only be calculated once at the beginning of the computation.

3.3 OBTAINING AND DISCRETIZING THE INTEGRAL FORM FOR A SCALAR PROBLEM: A CHEMICAL DIFFUSION EXAMPLE

Having presented the general principles of the geometric discretization into finite elements and the integration and derivation operations, we advance to the practical application for the solution of partial differential equations. To illustrate the method, we again take the example of stationary chemical diffusion with transport as seen already in chapter 2.

The unknown field is the scalar solute concentration, $c(\boldsymbol{x})$, in the domain Ω. The governing equation for the phenomenon in the absence of a solute source term, and in a stationary incompressible flow (equation (1.74), with $\partial c/\partial t = 0$) is:

$$\boldsymbol{v} \cdot \mathbf{grad}\, c - \mathrm{div}(D\, \mathbf{grad}\, c) = 0 \tag{3.52}$$

where $\boldsymbol{v}$ is the transport velocity vector, assumed uniform, and D the diffusion coefficient. In this example, we only consider two types of boundary conditions (fig. 3.13):

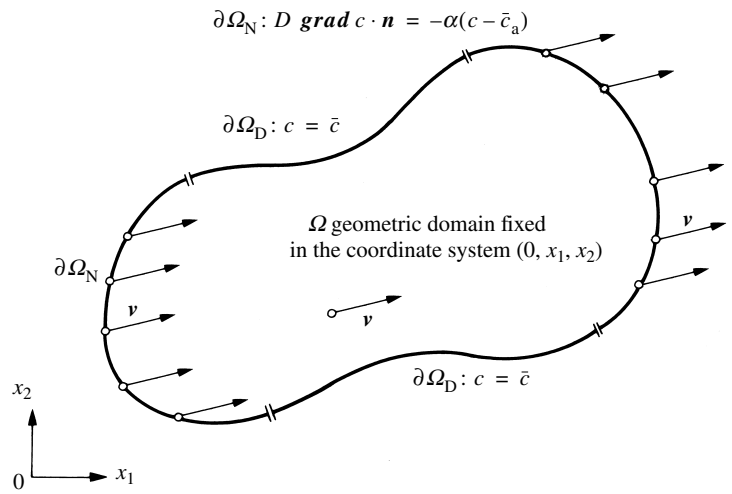


Figure 3.13 Boundary conditions for a chemical diffusion problem on a geometric domain Ω, in two dimensions and stationary in time. The velocity field $\boldsymbol{v}$ is assumed given and uniform.

- A Cauchy, or mixed, condition:

$$D\ \mathbf{grad}\ c \cdot \boldsymbol{n} = -\alpha(c - \bar{c}_a) \qquad \text{on } \partial\Omega_N \tag{3.53}$$

  where α is a chemical transfer coefficient with the ambient environment, where the concentration is $\bar{c}_a$.
- A Dirichlet condition (imposed concentration):

$$c = \bar{c} \qquad \text{on } \partial\Omega_D \tag{3.54}$$

3.3.1 Obtaining the integral form: weighted residuals method

We call a ***residual*** the value, scalar in this case, defined by:

$$R(c) = \boldsymbol{v} \cdot \mathbf{grad}\, c - \mathrm{div}(D\ \mathbf{grad}\, c) \tag{3.55}$$

The solution of the problem is clearly that which zeros the residual while simultaneously satisfying the boundary conditions at $\partial\Omega$. The basic idea of the solution method is to search for scalar functions c which zero the following integral form:

$$\Phi(c) = \int_\Omega \varphi R(c)\ \mathrm{d}V = \int_\Omega \varphi[\boldsymbol{v} \cdot \mathbf{grad}\, c - \mathrm{div}(D\ \mathbf{grad}\, c)]\ \mathrm{d}V = 0 \tag{3.56}$$

for every ***weighting function*** (or ***test function***) φ belonging to a set of functions E_φ, while c satisfies the boundary conditions at $\partial\Omega$.

The equivalence between $R(c) = 0$ on Ω and (3.56) is in fact only true if the set E_φ has infinite dimension and is composed of independent functions (for example, the set of Dirac delta functions $\delta(\boldsymbol{x})$ for every point $\boldsymbol{x}$ in Ω). If that is not the case, that is if E_φ is finite, (as is the case for the finite element method), the solution c which satisfies (3.56) is only an *approximate* solution to the problem.

In order to integrate (3.56), we can also write it in the form:

$$\int_\Omega \varphi \boldsymbol{v} \cdot \mathbf{grad}\, c \, \mathrm{d}V - \int_\Omega \mathrm{div}(\varphi D \, \mathbf{grad}\, c) \, \mathrm{d}V + \int_\Omega D \, \mathbf{grad}\, \varphi \cdot \mathbf{grad}\, c \, \mathrm{d}V = 0 \tag{3.57}$$

The divergence theorem then permits the transformation of the second term to obtain:

$$\int_\Omega \varphi \boldsymbol{v} \cdot \mathbf{grad}\, c \, \mathrm{d}V - \int_{\partial\Omega} \varphi D \, \mathbf{grad}\, c \cdot \boldsymbol{n} \, \mathrm{d}S + \int_\Omega D \, \mathbf{grad}\, \varphi \cdot \mathbf{grad}\, c \, \mathrm{d}V = 0 \tag{3.58}$$

It should be noted at this stage that this operation has allowed the reduction of the order of derivation of c (only first derivatives are needed, and no longer the second derivatives). The other advantage resulting from this formulation is that the boundary conditions can now be taken into account very easily. In fact, the integral on the boundary can be decomposed into two sums over $\partial\Omega_\mathrm{N}$ and $\partial\Omega_\mathrm{D}$. The concentration c being known (given) on $\partial\Omega_\mathrm{D}$, we can restrict the space E_φ of test functions to those that are zero on $\partial\Omega_\mathrm{D}$: $E_{\varphi 0} = \{ \varphi \in E_\varphi | \varphi = 0 \text{ on } \partial\Omega_\mathrm{D} \}$ (the reader is referred to the bibliography for a complete justification of this step). Using the boundary condition (3.53), the ***integral*** or ***weak form***, of the posed problem is obtained:

$$\int_\Omega \varphi \boldsymbol{v} \cdot \mathbf{grad}\, c \, \mathrm{d}V + \int_{\partial\Omega_\mathrm{N}} \varphi \alpha (c - \bar{c}_\mathrm{a}) \, \mathrm{d}S + \int_\Omega D \, \mathbf{grad}\, \varphi \cdot \mathbf{grad}\, c \, \mathrm{d}V = 0$$
$$\forall \varphi \in E_{\varphi 0} \tag{3.59}$$

3.3.2 Discretized integral form. Galerkin method

In order to concretely solve the problem and calculate c on the domain Ω, it will be necessary to include the spatial discretization into finite elements and to choose an ensemble of test functions E_φ. For the first point, designating by c_h the discrete approximation of the continuous field c, we can write:

$$c_\mathrm{h}(\boldsymbol{x}) = c^\mathrm{n} \psi_\mathrm{n}(\boldsymbol{x}) \tag{3.60}$$

the sum being over the *Nn* nodes of the grid. In the rest of the chapter we omit the index *h*, to simplify the expression. As for the discretization of the integral form (3.59), it is obtained by choosing *Nn* independent weighting functions (as many as the number of parameters defining the approximation c_h): φ_1, φ_2, ..., φ_{Nn}. The choice of the type of functions φ_i leads to different methods: collocation by points, by sub domains and Galerkin. We only present the latter here, which is by far the most used.

The ***Galerkin method*** consists of taking precisely the interpolation functions ψ_n of the approximation by finite elements for weighting functions. Thus if we *discretize* the weak form (3.59), applying the interpolation (3.60) and choosing $\varphi_i = \psi_i$, we obtain:

$$\int_\Omega \psi_i \boldsymbol{v} \cdot \mathbf{grad}(\psi_j c^j)\, dV + \int_{\partial\Omega_N} \alpha \psi_i \psi_j c^j\, dS + \int_\Omega D\, \mathbf{grad}(\psi_j c^j) \cdot \mathbf{grad}\,\psi_i\, dV = \int_{\partial\Omega_N} \alpha \bar{c}_a \psi_i\, dS \tag{3.61}$$

for the nodes *i* for which the value c^i is not imposed. We denote the set of these degrees of freedom N^{free}, while the set of degrees of freedom for the imposed values is denoted N^{imp}.

By virtue of the linear properties of the sums and integrals, and the boundary conditions (3.54), the discrete problem to solve is then written:

$$c^j \left[\int_\Omega \psi_i \boldsymbol{v} \cdot \mathbf{grad}\,\psi_j\, dV + \int_{\partial\Omega_N} \alpha \psi_i \psi_j\, dS + \int_\Omega D\, \mathbf{grad}\,\psi_j \cdot \mathbf{grad}\,\psi_i\, dV \right] = \int_{\partial\Omega_N} \alpha \bar{c}_a \psi_i\, dS \qquad \text{for } i \in N^{free} \tag{3.62}$$

and

$$c^i = \bar{c} \qquad \text{for } i \in N^{imp} \tag{3.63}$$

The concentration field under consideration now appears as the solution of a system of *Nn* equations and *Nn* unknowns:

$$[\boldsymbol{K}]\, \boldsymbol{c} = \boldsymbol{b} \tag{3.64}$$

in which $\boldsymbol{c}$ is the column vector having the nodal concentrations c^j, $j = 1, Nn$ for components.

The matrix $[\boldsymbol{K}]$ includes the diffusivity and advection terms. Its size is $Nn \times Nn$ and its components are given by:

$$K_{ij} = \int_{\Omega} \left[\psi_i v_k \frac{\partial \psi_j}{\partial x_k} + D \frac{\partial \psi_i}{\partial x_k} \frac{\partial \psi_j}{\partial x_k} \right] dV + \int_{\partial\Omega_N} \alpha \psi_i \psi_j \, dS \qquad \text{for } i \in N^{free} \quad (3.65)$$

and

$$K_{ij} = \delta_{ij} \qquad \text{for } i \in N^{imp} \quad (3.66)$$

k being summed from 1 to d, the spatial dimension of the problem. The generic name ***stiffness matrix***, whose origin is in the solution of mechanical equilibrium problems (sect. 3.4), is often used for the matrix $[\boldsymbol{K}]$. One can see in the present case that $[\boldsymbol{K}]$ is not symmetric, because of the term associated with advective transport. We will return to this subject in paragraph 3.6.5 for the solution of a non stationary problem; the specific treatment of the advective transport is detailed in chapter 7.

Finally, $\boldsymbol{b}$ is the right hand side vector defined by:

$$b_i = \int_{\partial\Omega_N} \alpha \bar{c}_a \psi_i \, dS \qquad \text{for } i \in N^{free} \quad (3.67)$$

and

$$b_i = \bar{c} \qquad \text{for } i \in N^{imp} \quad (3.68)$$

The method of calculation of $[\boldsymbol{K}]$ and $\boldsymbol{b}$, by decomposition for each of the elements and assembly, will be detailed in section 3.5.2. Note that if the diffusion coefficient D does not depend on the concentration c, the matrix $[\boldsymbol{K}]$ of the system will not depend on the unknown vector $\boldsymbol{c}$ and the system is therefore linear. Otherwise, the system is non linear. The techniques for solution of the system in the two cases are explained in chapter 4.

The diffusion part of matrix $[\boldsymbol{K}]$ has an interesting physical interpretation in the case of a linear triangular element: it is shown in chapter 10 that in each element, the terms of this matrix correspond to the solute flux exchanged across the medians of the triangles between two nodes.

3.3.3 The least square method

The weighted residual method is not the only one that can be used to search for a function that zeros the residual $R(c)$ on Ω. The principle of least squares, for example, consists of searching for functions c that minimize the following integral:

$$f(c) = \int_{\Omega} (R(c))^2 \, dV \quad (3.69)$$

and that respect the conditions at the boundary of Ω. This method, natural enough in appearance, suffers two major inconveniences. First, the boundary conditions are

difficult to take into account, except for Dirichlet conditions (that is, $c = \bar{c}$ on $\partial\Omega = \partial\Omega_D$). Second, the order of the derivatives in R can not be reduced: in practice, this leads to high differentiability conditions on the finite element discretization. For these two reasons the method of weighted residuals discussed above is preferred.

3.3.4 Matrix presentation

We will now describe the terms appearing in (3.62) explicitly in terms of matrices. For that we denote by Ψ the line matrix composed of the Nn interpolation functions:

$$\Psi = [\psi_1(\boldsymbol{x})\ \psi_2(\boldsymbol{x}) \dots \psi_{\mathrm{Nn}}(\boldsymbol{x})] \tag{3.70}$$

For every point $\boldsymbol{x}$, the fundamental relation for interpolation (3.60) can now be written in the form of a product:

$$c(\boldsymbol{x}) = [\psi_1(\boldsymbol{x})\psi_2(\boldsymbol{x})\dots\psi_{\mathrm{Nn}}(\boldsymbol{x})] \begin{bmatrix} c^1 \\ c^2 \\ \vdots \\ c^{\mathrm{Nn}} \end{bmatrix} = \Psi \boldsymbol{c} \tag{3.71}$$

Similarly, for each element e containing nne nodes, we can write:

$$c(\boldsymbol{x}) = [\psi_1^{\mathrm{e}}(\boldsymbol{x})\psi_2^{\mathrm{e}}(\boldsymbol{x})\dots\psi_{\mathrm{nne}}^{\mathrm{e}}(\boldsymbol{x})] \begin{bmatrix} c^1 \\ c^2 \\ \vdots \\ c^{\mathrm{nne}} \end{bmatrix} = \Psi^{\mathrm{e}} \boldsymbol{c}^{\mathrm{e}} \tag{3.72}$$

Suppose that we want to calculate the gradient vector of the field $c(\boldsymbol{x})$. In two dimensions, this takes the form:

$$\mathbf{grad}\, c = \begin{bmatrix} \dfrac{\partial c}{\partial x_1} \\ \dfrac{\partial c}{\partial x_2} \end{bmatrix} \tag{3.73}$$

Taking into account that the derivatives of c are expressed in terms of the finite element interpolation by:

$$\frac{\partial c(\boldsymbol{x})}{\partial x_{\mathrm{k}}} = \frac{\partial}{\partial x_{\mathrm{k}}}(\psi_{\mathrm{p}}(\boldsymbol{x})c^{\mathrm{p}}) = \frac{\partial \psi_{\mathrm{p}}(\boldsymbol{x})}{\partial x_{\mathrm{k}}}c^{\mathrm{p}} \tag{3.74}$$

the gradient of the concentration field can be expressed, for every point in space, as a matrix product:

$$\mathbf{grad}\, c = \begin{bmatrix} \dfrac{\partial \psi_1}{\partial x_1} & \dfrac{\partial \psi_2}{\partial x_1} & \cdots & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_1} \\ \dfrac{\partial \psi_1}{\partial x_2} & \dfrac{\partial \psi_2}{\partial x_2} & \cdots & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_2} \end{bmatrix} \begin{bmatrix} c^1 \\ c^2 \\ \vdots \\ c^{\mathrm{Nn}} \end{bmatrix} = [\boldsymbol{B}]\boldsymbol{c} \tag{3.75}$$

Introducing the derivative column vector $\boldsymbol{\nabla}$ acting on the components of the line vector Ψ (in 2 dimensions):

$$\boldsymbol{\nabla} = \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \end{bmatrix} \tag{3.76}$$

the matrix equation obtained is:

$$\mathbf{grad}\, c = \boldsymbol{\nabla} c = \boldsymbol{\nabla}\Psi(\boldsymbol{x})\boldsymbol{c} = [\boldsymbol{B}(\boldsymbol{x})]\boldsymbol{c} \tag{3.77}$$

As for the matrix expression of the weak form resulting from the Galerkin method (paragraph 3.3.2), it can be verified without difficulty that introducing the notions above in the system matrix and its right hand side vector (3.65) and (3.67), they become as follows:

$$[\boldsymbol{K}] = \int_{\Omega} [D[\boldsymbol{B}]^{\mathrm{T}}[\boldsymbol{B}] + \Psi^{\mathrm{T}}[\boldsymbol{B}]^{\mathrm{T}}\boldsymbol{v}]\, \mathrm{d}V + \int_{\partial\Omega_{\mathrm{N}}} \alpha \Psi^{\mathrm{T}}\Psi\, \mathrm{d}S \tag{3.78}$$

$$\boldsymbol{b} = \int_{\partial\Omega_{\mathrm{N}}} \alpha \bar{c}_{\mathrm{a}} \Psi^{\mathrm{T}}\, \mathrm{d}S \tag{3.79}$$

3.4 SOLUTION OF A VECTOR PROBLEM: MECHANICAL EQUILIBRIUM EXAMPLE

The objective of this section is to generalize the FEM formalism to problems in which the unknown is a vector field with several degrees of freedom per node, taking as an example the components of velocity or displacement in mechanics. In this

case the starting point for the application of the FEM will be the principle of virtual power. We will analyze in detail a case of linear elasticity for which there exists a functional whose minimum gives the equilibrium solution, in a manner similar to the principle of virtual power.

3.4.1 Equilibrium equation and boundary conditions

In chapter 1 it was seen that in the absence of inertia and volume density forces, the local equilibrium equation is,

$$\mathbf{div}\ \boldsymbol{\sigma} = \mathbf{0} \tag{3.80}$$

where $\boldsymbol{\sigma}$ is the stress tensor. From the point of view of the application of the FEM, it is preferable to employ the equivalent equation, expressed as virtual power, as it is already in integral form. With the same assumptions, (1.49) becomes:

$$\int_{\Omega} \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{*}\, \mathrm{d}V = \int_{\partial\Omega} \boldsymbol{T} \cdot \boldsymbol{v}^{*}\, \mathrm{d}S \qquad \text{for all } \boldsymbol{v}^{*} \tag{3.81}$$

or, employing the symmetry of the tensor $\boldsymbol{\sigma}$ and the relation $\dot{\boldsymbol{\varepsilon}}^{*} = 1/2[\mathbf{Grad}\,\boldsymbol{v}^{*} + (\mathbf{Grad}\,\boldsymbol{v}^{*})^{\mathrm{T}}]$,

$$\int_{\Omega} \boldsymbol{\sigma} : \mathbf{Grad}\,\boldsymbol{v}^{*}\, \mathrm{d}V = \int_{\partial\Omega} \boldsymbol{T} \cdot \boldsymbol{v}^{*}\, \mathrm{d}S \qquad \text{for all } \boldsymbol{v}^{*} \tag{3.82}$$

where $\dot{\boldsymbol{\varepsilon}}^{*}$ is the tensor of virtual strain rates (associated with the virtual velocity field $\boldsymbol{v}^{*}$), $\boldsymbol{T}$ is the stress vector on the surface of Ω ($\boldsymbol{T} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$, $\boldsymbol{n}$ being the outgoing normal vector).

The ***boundary conditions*** are principally of two types:

Imposed velocity vector (Dirichlet condition): $\boldsymbol{v} = \bar{\boldsymbol{v}}$ on $\partial\Omega_{\mathrm{D}}$ (3.83)

Imposed stress vector: $\boldsymbol{T} = \overline{\boldsymbol{T}}$ on $\partial\Omega_{\mathrm{N}}$ (3.84)

In fact, as can be seen in figure 3.14, these conditions are only rough approximations of the boundary conditions encountered in practice. In reality, the conditions are imposed *component by component* on material points.

Thus one has:

$v_{\mathrm{k}} = \bar{v}_{\mathrm{k}}$ in the x_{k} direction. (3.85)

Such is the case for the normal direction in figure 3.14, where the component of imposed velocity is that of the tool.

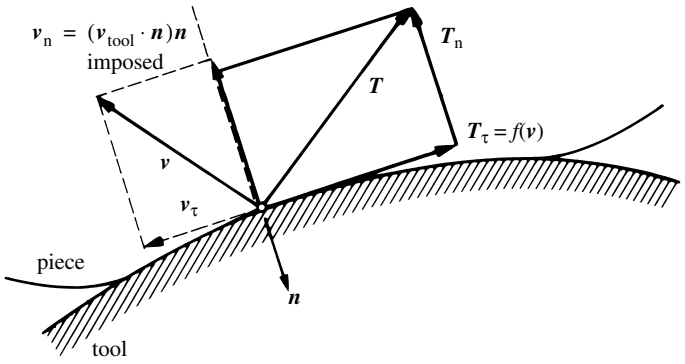


Figure 3.14 Boundary conditions for a problem in mechanics.

For the components where the velocity is not imposed, a choice exists:

$$\text{either } T_k = \bar{T}_k \text{ or } T_k = f(\boldsymbol{v}) \tag{3.86}$$

where $f(\boldsymbol{v})$ is a function of the velocity vector, which is unknown. Such is the case for the tangential component in figure 3.14, $f(\boldsymbol{v})$ coming from a friction law at the tool surface. The reader is referred to chapter 6 for a more detailed discussion of this point, as well as for a treatment of the rotations needed for the expression of the boundary conditions, as the coordinates x_k do not necessarily correspond to the laboratory coordinates.

For the continuous formulation, it is preferable to maintain the notations $\partial\Omega_D$ and $\partial\Omega_N$, to avoid needlessly encumbering the expressions. This choice has no effect on the results. However, we will see that a clarification is possible in the expression of the discretized formulation.

3.4.2 Linear elasticity example. Existence of a functional

Hooke's law for the behavior of a solid, characteristic of ***linear elasticity***, links the stress tensor $\boldsymbol{\sigma}$ to the strain tensor $\boldsymbol{\varepsilon}$ (chap. 1 equation(1.87)):

$$\sigma_{ij} = \frac{E}{1+\nu_P}\varepsilon_{ij} + \frac{E\nu_P}{(1+\nu_P)(1-2\nu_P)}\mathrm{Tr}(\boldsymbol{\varepsilon})\delta_{ij} \tag{3.87}$$

where E is Young's modulus and ν_P is Poisson's coefficient. The equation can also be expressed in tensor form:

$$\boldsymbol{\sigma} = \boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} \tag{3.88}$$

where $\boldsymbol{D}^{e\ell}$ is the elasticity tensor, of 4^{th} order. In d dimensions, this equation can be written for the components as follows:

$$\sigma_{ij} = D^{e\ell}_{ijkl}\varepsilon_{kl} \quad \text{with} \quad D^{e\ell}_{ijkl} = \frac{E}{1+\nu_P}\delta_{ik}\delta_{jl} + \frac{E\nu_P}{(1+\nu_P)(1-2\nu_P)}\delta_{ij}\delta_{kl}$$
for $i, j, k, l = 1, d$ (3.89)

As the constitutive law links the stresses to the strains, and thus to displacements, it is preferable to formulate the problem with the ***principle of virtual work*** which is strictly equivalent to that for virtual power (3.81), but which replaces the velocity field $\boldsymbol{v}^*$ by a field of infinitesimal displacements $\boldsymbol{\delta u}^*$; which, denoting $\boldsymbol{\delta\varepsilon}^*$ the strain tensor associated with $\boldsymbol{\delta u}^*$, leads to:

$$\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}^* \, dV = \int_{\partial\Omega} \boldsymbol{T} \cdot \delta\boldsymbol{u}^* \, dS \qquad \text{for all } \delta\boldsymbol{u}^* \tag{3.90}$$

If $\boldsymbol{u}$ is the vector displacement for every point with respect to its initial position in Ω, without internal stress, and if $\boldsymbol{\varepsilon}$ is the strain tensor associated with $\boldsymbol{u}$ at every point, then the principle of virtual work (3.90) takes the form:

$$\int_\Omega \boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon}^* \, dV = \int_{\partial\Omega} \boldsymbol{T} \cdot \delta\boldsymbol{u}^* \, dS \qquad \text{for all } \delta\boldsymbol{u}^* \tag{3.91}$$

If we consider $\boldsymbol{\delta u}^*$ to be the variation $\boldsymbol{\delta u}$ of the real displacements $\boldsymbol{u}$, then the scalar value $\boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon}$ appears as the variation of the term $1/2\boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}$. The verification is simple, based on the symmetry (ij, kl) of the elasticity tensor $\boldsymbol{D}^{e\ell}$. We have:

$$\boldsymbol{D}^{e\ell}\delta\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} = D^{e\ell}_{ijkl}\delta\varepsilon_{kl}\varepsilon_{ij} = D^{e\ell}_{klij}\delta\varepsilon_{kl}\varepsilon_{ij} = D^{e\ell}_{klij}\varepsilon_{ij}\delta\varepsilon_{kl} = \boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon} \tag{3.92}$$

and thus:

$$\delta\left(\tfrac{1}{2}\boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}\right) = \tfrac{1}{2}\boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon} + \tfrac{1}{2}\boldsymbol{D}^{e\ell}\delta\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} = \boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon} \tag{3.93}$$

Equation (3.91) appears in these conditions as a stationary condition for the quantity W defined as:

$$W = \int_\Omega \tfrac{1}{2}\boldsymbol{D}^{e\ell}\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \, dV - \int_{\partial\Omega} \boldsymbol{T} \cdot \boldsymbol{u} \, dS \tag{3.94}$$

which is the ***elastic potential energy*** and which is minimized at equilibrium. We have thus seen two equivalent expressions of the equilibrium equation for an elastic medium: first the principle of virtual work or power, and second, the variational formulation which consists of minimizing a function for the elastic potential energy.

The variational principle or the minimization of a functional (sum of integral functions over the domain and, possibly, the boundary) is shared by many problems that are called *conservative*. We will see another example in chapter 6 (section 6.5.3): the viscoplastic functional.

3.4.3 Discretization

Now we introduce finite element discretization in the principles discussed above. For the virtual and real velocity fields we have:

$$\boldsymbol{v}^* = \psi_n \boldsymbol{v}^{*n} \quad \text{and} \quad \boldsymbol{v} = \psi_n \boldsymbol{v}^n \tag{3.95}$$

where n varies from 1 to Nn, the number of nodes. The components of the tensor velocity gradient (virtual or real) can be easily expressed as functions of the components of the nodal velocity field:

$$\frac{\partial v_i^*}{\partial x_j} = \frac{\partial}{\partial x_j}(\psi_n v_i^{*n}) = \frac{\partial \psi_n}{\partial x_j} v_i^{*n} \tag{3.96}$$

where i and j vary from 1 to d, the spatial dimension of the problem. In general the components (i, j) of the strain rate tensor associated with $\boldsymbol{v}^*$ can be linked to the components (n, k) of the velocity field:

$$\dot{\varepsilon}_{ij}^* = \frac{1}{2}\left(\frac{\partial v_i^*}{\partial x_j} + \frac{\partial v_j^*}{\partial x_i}\right) = B_{ijnk} v_k^{*n} \tag{3.97}$$

where k varies from 1 to N, the number of unknowns per node[1]. The coefficients B_{ijnk} are given by:

$$B_{ijnk} = \frac{1}{2}\left(\frac{\partial \psi_n}{\partial x_j}\delta_{ik} + \frac{\partial \psi_n}{\partial x_i}\delta_{jk}\right) \tag{3.98}$$

Applying this discretization to the principle of virtual power (3.81):

$$\int_\Omega \sigma_{ij} B_{ijnk} v_k^{*n} \, dV = \int_{\partial\Omega} T_k \psi_n v_k^{*n} \, dS \quad \text{for all } v_k^{*n} \tag{3.99}$$

As the values v_k^{*n} can be chosen arbitrarily, the following can be deduced,

$$\int_\Omega \sigma_{ij} B_{ijnk} \, dV = \int_{\partial\Omega} T_k \psi_n \, dS \quad \text{for all } (n, k)\,. \tag{3.100}$$

[1] Most often, $N = d$ (2 unknown velocity components in 2 dimensions, 3 in 3 dimensions), but it is not always true: for example, there can be three unknowns v_r, v_θ, v_z for a problem posed in a two dimensional section (r, z) of a domain.

The choice of the spatial discretization (3.95a) corresponds to a Galerkin formulation. As an exercise, the reader can verify that the result (3.100) is equivalent to a Galerkin formulation applied to each component of the local equilibrium equation **div** $\boldsymbol{\sigma} = \mathbf{0}$:

$$\int_{\Omega} \psi_{\mathrm{n}} \frac{\partial \sigma_{\mathrm{kj}}}{\partial x_{\mathrm{j}}} \, \mathrm{d}V = 0 \qquad \text{for all } (n, k) \tag{3.101}$$

Discretization of the boundary conditions. Nodal reactions

The imposition of velocity components for some nodes on the boundary $\partial\Omega$ (3.83) is expressed after discretization by:

$$v_{\mathrm{k}}^{\mathrm{n}} = \bar{v}_{\mathrm{k}}^{\mathrm{n}} \tag{3.102}$$

for certain degrees of freedom (n, k). We distinguish thus between "imposed" degrees of freedom (abbreviated *dof*) and "free", using again the notation N^{free} and N^{imp} introduced in paragraph 3.3.2 for the sets of *dof* which are "free" and "imposed", respectively. For these conditions, we can choose a discretized virtual velocity field which is zero for the "imposed" *dof*, that is, which satisfies:

$$v_{\mathrm{k}}^{*\mathrm{n}} = 0 \qquad \text{for } (n, k) \in N^{\mathrm{imp}} \tag{3.103}$$

Such a field is called ***kinematically admissible to zero***. Thus one must solve a system of (NnN) equations with as many unknowns:

$$\int_{\Omega} \sigma_{\mathrm{ij}} B_{\mathrm{ijnk}} \, \mathrm{d}V = \int_{\partial\Omega} T_{\mathrm{k}} \psi_{\mathrm{n}} \, \mathrm{d}S \qquad \text{for } (n, k) \in N^{\mathrm{free}} \tag{3.104}$$

and

$$v_{\mathrm{k}}^{\mathrm{n}} = \bar{v}_{\mathrm{k}}^{\mathrm{n}} \qquad \text{for } (n, k) \in N^{\mathrm{imp}} \tag{3.105}$$

Imagine now *the same mechanical equilibrium*, but this time obtained with *nodal forces* $\bar{\boldsymbol{F}}$ acting in the direction of the "imposed" *dof*, instead of the previously imposed velocity conditions (fig. 3.15).

The principle of virtual power in this case is written in continuous form:

$$\int_{\Omega} \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{*} \, \mathrm{d}V - \int_{\partial\Omega} \boldsymbol{T} \cdot \boldsymbol{v}^{*} \, \mathrm{d}S - \sum_{\substack{\text{points M} \\ \text{with imposed } dof}} \bar{\boldsymbol{F}}^{\mathrm{M}} \cdot \boldsymbol{v}^{*\mathrm{M}} = 0 \tag{3.106}$$

and in discrete form:

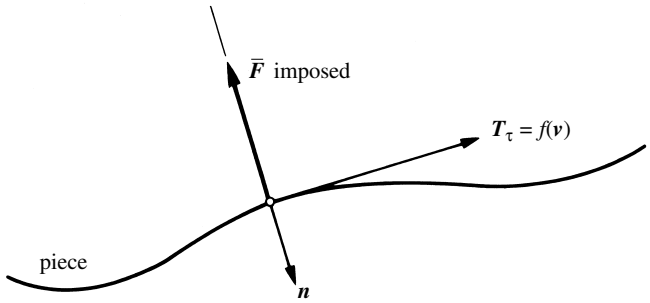


Figure 3.15 Applied normal force corresponding to the imposition of a perpendicular degree of freedom of the velocity field in figure 3.14.

$$\int_\Omega \sigma_{ij} B_{ijnk}\, dV - \int_{\partial\Omega} T_k \psi_n\, dS = 0 \qquad \text{for } (n,k) \in N^{free} \tag{3.107}$$

$$\int_\Omega \sigma_{ij} B_{ijnk}\, dV - \int_{\partial\Omega} T_k \psi_n\, dS - \bar{F}^n_k = 0 \qquad \text{for } (n,k) \in N^{imp} \tag{3.108}$$

We then conclude from (3.108) that the components of the *applied nodal forces* (the ***nodal reactions***) associated with the imposed degrees of freedom of velocity are in fact equal to the *residuals of the equilibrium equation according to these dofs*:

$$F^n_k = \int_\Omega \sigma_{ij} B_{ijnk}\, dV - \int_{\partial\Omega} T_k \psi_n\, dS \tag{3.109}$$

Linear elasticity example

In the linear elastic case, the stresses are expressed with the constitutive law (3.89) :

$$\sigma_{ij} = D^{e\ell}_{ijpq} \varepsilon_{pq} \tag{3.110}$$

In the same way as for (3.97), the components ε_{pq} of the strain tensor can be expressed as a function of the nodal components of the displacement field, u^m_l, with the coefficients B_{pqml}:

$$\sigma_{ij} = D^{e\ell}_{ijpq} B_{pqml} u^m_l \tag{3.111}$$

Finally, we have a system of (NnN) equations to solve with as many unknowns:

$$\int_{\Omega} B_{\mathrm{ijnk}} D^{\mathrm{e}\ell}_{\mathrm{ijpq}} B_{\mathrm{pqml}} u^{\mathrm{m}}_{\mathrm{l}} \, \mathrm{d}V - \int_{\partial\Omega} T_{\mathrm{k}} \psi_{\mathrm{n}} \, \mathrm{d}S = 0 \qquad \text{for } (n, k) \in N^{\mathrm{free}} \tag{3.112}$$

and

$$u^{\mathrm{n}}_{\mathrm{k}} = \bar{u}^{\mathrm{n}}_{\mathrm{k}} \qquad \text{for } (n, k) \in N^{\mathrm{imp}} \tag{3.113}$$

As long as the elastic coefficients are constant and the stress vectors fixed (or can be expressed as linear functions of the displacement field), these equations are linear in the nodal unknowns $u^{\mathrm{m}}_{\mathrm{l}}$. The system can then be written:

$$K_{\mathrm{nkml}} u^{\mathrm{m}}_{\mathrm{l}} = F_{\mathrm{nk}} \qquad n, m = 1, Nn \qquad \text{and} \qquad k, l = 1, N \tag{3.114}$$

The system matrix, [$\boldsymbol{K}$] – the *stiffness matrix* – has dimensions (NnN) × (NnN). When the stress vectors are fixed, the components (nk, ml) of the matrix are given by:

$$K_{\mathrm{nkml}} = \int_{\Omega} B_{\mathrm{ijnk}} D^{\mathrm{e}\ell}_{\mathrm{ijpq}} B_{\mathrm{pqml}} \, \mathrm{d}V \qquad \text{for } (n, k) \in N^{\mathrm{free}} \tag{3.115}$$

$$K_{\mathrm{nkml}} = \delta_{\mathrm{nm}} \delta_{\mathrm{kl}} \qquad \text{for } (n, k) \in N^{\mathrm{imp}} \tag{3.116}$$

The term on the right hand side, $\boldsymbol{F}$, is a vector of dimension (NnN), for which the components (nk) are given by:

$$F_{\mathrm{nk}} = \int_{\partial\Omega} T_{\mathrm{k}} \psi_{\mathrm{n}} \, \mathrm{d}S \qquad \text{for } (n, k) \in N^{\mathrm{free}} \tag{3.117}$$

$$F_{\mathrm{nk}} = \bar{u}^{\mathrm{n}}_{\mathrm{k}} \qquad \text{for } (n, k) \in N^{\mathrm{imp}} \tag{3.118}$$

It can be verified that the minimization of the functional W (3.94) with respect to the nodal unknowns, that is writing $\partial W / \partial u^{\mathrm{n}}_{\mathrm{k}} = 0$ for every pair (n, k), leads to exactly the same result.

Lastly, it is seen that the system matrix, [$\boldsymbol{K}$], is symmetric (nk, ml). It must be noted in this context that the minimization of a functional always leads to the resolution of a symmetric system, unlike the application of the Galerkin method; which in the case where one can not define a functional, can lead to non symmetric systems such as (3.64) for example.

3.4.4 Matrix Presentation

Take the example of a two dimensional mechanics calculation (the extension to three dimensions causes no specific difficulties) in which we determine the two components of the displacement vector $\boldsymbol{u}$ at each node of the finite element discretization.

We denote by $\boldsymbol{U}$ the vector of the nodal components of the field $\boldsymbol{u}(\boldsymbol{x})$.

$$\boldsymbol{U} = \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ \vdots \\ u_1^{\mathrm{Nn}} \\ u_2^{\mathrm{Nn}} \end{bmatrix} \tag{3.119}$$

We have for each of the components $u_1(\boldsymbol{x})$ and $u_2(\boldsymbol{x})$ the fundamental relation of interpolation (3.1), in matrix form:

$$\begin{bmatrix} u_1(\boldsymbol{x}) \\ u_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \psi_1(\boldsymbol{x}) & 0 & \psi_2(\boldsymbol{x}) & 0 & \dots & \psi_{\mathrm{Nn}}(\boldsymbol{x}) & 0 \\ 0 & \psi_1(\boldsymbol{x}) & 0 & \psi_2(\boldsymbol{x}) & \dots & 0 & \psi_{\mathrm{Nn}}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ \vdots \\ u_1^{\mathrm{Nn}} \\ u_2^{\mathrm{Nn}} \end{bmatrix} \tag{3.120}$$

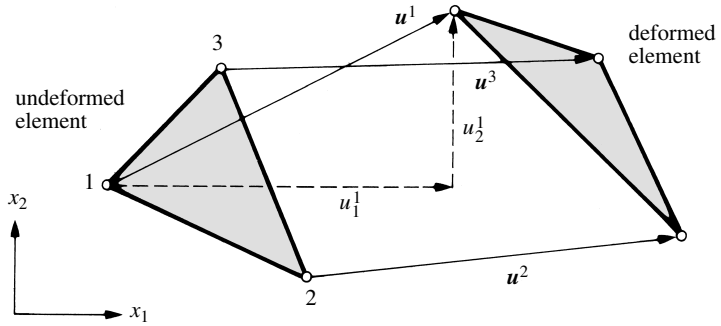


Figure 3.16 Displacements of the nodes of a linear triangular finite element.

or further condensed,

$$\boldsymbol{u}(\boldsymbol{x}) = [\boldsymbol{\Psi}(\boldsymbol{x})]\boldsymbol{U} \tag{3.121}$$

Note that this time $[\boldsymbol{\Psi}]$ is a matrix of dimension $2 \times (2Nn)$, or, generally, $N \times (NNn)$. At the level of each element, we have similarly:

$$\boldsymbol{u} = \begin{bmatrix} u_1(\boldsymbol{x}) \\ u_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \psi_1^{\mathrm{e}}(\boldsymbol{x}) & 0 & \psi_2^{\mathrm{e}}(\boldsymbol{x}) & 0 & \dots & \psi_{\mathrm{nne}}^{\mathrm{e}}(\boldsymbol{x}) & 0 \\ 0 & \psi_1^{\mathrm{e}}(\boldsymbol{x}) & 0 & \psi_2^{\mathrm{e}}(\boldsymbol{x}) & \dots & 0 & \psi_{\mathrm{nne}}^{\mathrm{e}}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ \vdots \\ u_1^{\mathrm{nne}} \\ u_2^{\mathrm{nne}} \end{bmatrix}$$

$$= [\boldsymbol{\Psi}^{\mathrm{e}}]\boldsymbol{U}^{\mathrm{e}} \tag{3.122}$$

Suppose now that from the displacement field we wish to calculate the components of the strain tensor, that we give here in vector form:

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial u_1}{\partial x_1} \\ \dfrac{\partial u_2}{\partial x_2} \\ \dfrac{\partial u_1}{\partial x_2} + \dfrac{\partial u_2}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \varepsilon_{11}(\boldsymbol{x}) \\ \varepsilon_{22}(\boldsymbol{x}) \\ 2\varepsilon_{12}(\boldsymbol{x}) \end{bmatrix} \tag{3.123}$$

The vector $\boldsymbol{\varepsilon}$ contains the three components of the strain tensor, except for an extra factor of 2 in the third component. Considering that the derivatives of $\boldsymbol{u}$ are expressed in terms of finite element interpolation:

$$\frac{\partial u_{\mathrm{i}}(\boldsymbol{x})}{\partial x_{\mathrm{k}}} = \frac{\partial}{\partial x_{\mathrm{k}}}(\psi_{\mathrm{p}}(\boldsymbol{x})u_{\mathrm{i}}^{\mathrm{p}}) = \frac{\partial \psi_{\mathrm{p}}(\boldsymbol{x})}{\partial x_{\mathrm{k}}}u_{\mathrm{i}}^{\mathrm{p}} \qquad (p = 1, Nn) \tag{3.124}$$

we introduce the matrix derivative operator $[\boldsymbol{\nabla}]$:

$$[\boldsymbol{\nabla}] = \begin{bmatrix} \dfrac{\partial}{\partial x_1} & 0 \\ 0 & \dfrac{\partial}{\partial x_2} \\ \dfrac{\partial}{\partial x_2} & \dfrac{\partial}{\partial x_1} \end{bmatrix} \tag{3.125}$$

In this manner, we have:

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = [\boldsymbol{\nabla}]\boldsymbol{u}(\boldsymbol{x}) = [\boldsymbol{\nabla}][\boldsymbol{\Psi}(\boldsymbol{x})]\boldsymbol{U} = [\boldsymbol{B}]\boldsymbol{U} \tag{3.126}$$

where the matrix $[\boldsymbol{B}]$, defined by the product $[\boldsymbol{\nabla}]\,[\boldsymbol{\Psi}]$, is expressed as follows.

$$[\boldsymbol{B}] = [\boldsymbol{\nabla}][\boldsymbol{\Psi}] = \begin{bmatrix} \dfrac{\partial \psi_1}{\partial x_1} & 0 & \dfrac{\partial \psi_2}{\partial x_1} & 0 & & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_1} & 0 \\ 0 & \dfrac{\partial \psi_1}{\partial x_2} & 0 & \dfrac{\partial \psi_2}{\partial x_2} & \cdots & 0 & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_2} \\ \dfrac{\partial \psi_1}{\partial x_2} & \dfrac{\partial \psi_1}{\partial x_1} & \dfrac{\partial \psi_2}{\partial x_2} & \dfrac{\partial \psi_2}{\partial x_1} & & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_2} & \dfrac{\partial \psi_{\mathrm{Nn}}}{\partial x_1} \end{bmatrix} \tag{3.127}$$

In an analogous manner, we can use formulas (3.123) through (3.126) to express the strain rate tensor $\dot{\boldsymbol{\varepsilon}}^*$ as a function of the discretized virtual velocity field, $\boldsymbol{v}^*$. Using the same vector notation,

$$\dot{\boldsymbol{\varepsilon}}^* = \begin{bmatrix} \dfrac{\partial v_1^*}{\partial x_1^*} \\ \dfrac{\partial v_2^*}{\partial x_2} \\ \dfrac{\partial v_1^*}{\partial x_2} + \dfrac{\partial v_2^*}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \dot{\varepsilon}_{11}^* \\ \dot{\varepsilon}_{22}^* \\ 2\dot{\varepsilon}_{12}^* \end{bmatrix} \tag{3.128}$$

and we have:

$$\dot{\boldsymbol{\varepsilon}}^* = [\boldsymbol{\nabla}]\boldsymbol{v}^* = [\boldsymbol{\nabla}][\boldsymbol{\Psi}]\boldsymbol{V}^* = [\boldsymbol{B}]\boldsymbol{V}^* \tag{3.129}$$

where $\boldsymbol{V}^*$ denotes the nodal virtual velocity vector. The vectorized form of the stress tensor is given by:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} \tag{3.130}$$

Thus the volumetric strain power associated with the virtual velocity field $\boldsymbol{v}^*$ is given by the scalar product of the column vectors $\dot{\boldsymbol{\varepsilon}}^*$ and $\boldsymbol{\sigma}$:

$$\sigma_{\mathrm{ij}}\dot{\varepsilon}_{\mathrm{ij}}^* = (\dot{\boldsymbol{\varepsilon}}^*)^{\mathrm{T}}\boldsymbol{\sigma} \quad (i, j = 1, 2) \tag{3.131}$$

In this way the matrix form of the virtual power principle (3.81) – considered for the case without mass forces or inertia – is the following:

$$\int_{\Omega} (\dot{\boldsymbol{\varepsilon}}^{*})^{\mathrm{T}} \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\partial\Omega} ([\boldsymbol{\Psi}]\boldsymbol{V}^{*})^{\mathrm{T}} \boldsymbol{T} \, \mathrm{d}S \qquad \text{for all } \boldsymbol{V}^{*} \tag{3.132}$$

Using (3.129), we obtain:

$$\int_{\Omega} \boldsymbol{V}^{*\mathrm{T}} [\boldsymbol{B}]^{\mathrm{T}} \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\partial\Omega} \boldsymbol{V}^{*\mathrm{T}} [\boldsymbol{\Psi}]^{\mathrm{T}} \boldsymbol{T} \, \mathrm{d}S \qquad \text{for all } \boldsymbol{V}^{*} \tag{3.133}$$

Because this relation must be satisfied for every field $\boldsymbol{V}^{*}$, we have finally:

$$\int_{\Omega} [\boldsymbol{B}]^{\mathrm{T}} \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\partial\Omega} [\boldsymbol{\Psi}]^{\mathrm{T}} \boldsymbol{T} \, \mathrm{d}S \tag{3.134}$$

Note that this equality is simply the discrete matrix expression of the virtual power principle, and that it is thus valid regardless of the constitutive law for the material under consideration. We will see the applications in chapter 6.

Application in the linear elastic case

If we continue now, the matrix expression of (3.87) in the context of linear elasticity and taking into account the vectorization of the stress tensor and the deformation tensor, is:

$$\boldsymbol{\sigma} = [\boldsymbol{D}^{\mathrm{e}\ell}] \boldsymbol{\varepsilon} \tag{3.135}$$

with

$$[\boldsymbol{D}^{\mathrm{e}\ell}] = \begin{bmatrix} \dfrac{E(1-\nu_{\mathrm{P}})}{(1+\nu_{\mathrm{P}})(1-2\nu_{\mathrm{P}})} & \dfrac{E\nu_{\mathrm{P}}}{(1+\nu_{\mathrm{P}})(1-2\nu_{\mathrm{P}})} & 0 \\ \dfrac{E\nu_{\mathrm{P}}}{(1+\nu_{\mathrm{P}})(1-2\nu_{\mathrm{P}})} & \dfrac{E(1-\nu_{\mathrm{P}})}{(1+\nu_{\mathrm{P}})(1-2\nu_{\mathrm{P}})} & 0 \\ 0 & 0 & \dfrac{E}{2(1+\nu_{\mathrm{P}})} \end{bmatrix} \tag{3.136}$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} \qquad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} \tag{3.137}$$

With the help of (3.126), (3.134) becomes:

$$\int_{\Omega} [\boldsymbol{B}]^{\mathrm{T}} [\boldsymbol{D}^{\mathrm{e}\ell}] [\boldsymbol{B}] \boldsymbol{U} \, \mathrm{d}V = \int_{\partial\Omega} [\boldsymbol{\Psi}]^{\mathrm{T}} \boldsymbol{T} \, \mathrm{d}S \tag{3.138}$$

The vector $\boldsymbol{U}$ of nodal displacements is then a solution of the linear system

$$[\boldsymbol{K}] \boldsymbol{U} = \boldsymbol{F} \tag{3.139}$$

in which the matrix and the right hand side term are given in matrix form below, and as components by equations (3.115) and (3.117).

$$[\boldsymbol{F}] = \int_{\Omega} [\boldsymbol{B}]^{\mathrm{T}} [\boldsymbol{D}^{\mathrm{e}\ell}] [\boldsymbol{B}] \, \mathrm{d}V \quad \text{and} \quad [\boldsymbol{F}] = \int_{\partial\Omega} [\boldsymbol{\Psi}]^{\mathrm{T}} \boldsymbol{T} \, \mathrm{d}S \tag{3.140}$$

Concluding this section on matrix notation, it should be said that it is of limited interest for the finite element program developer. In fact, the indexed notation is much closer to that which is actually programmed and leads to a more efficient vectorization. However, the matrix notation is useful for more theoretical studies of the convergence of the method, error estimation and the mesh adaptation.

3.5 IMPLEMENTATION

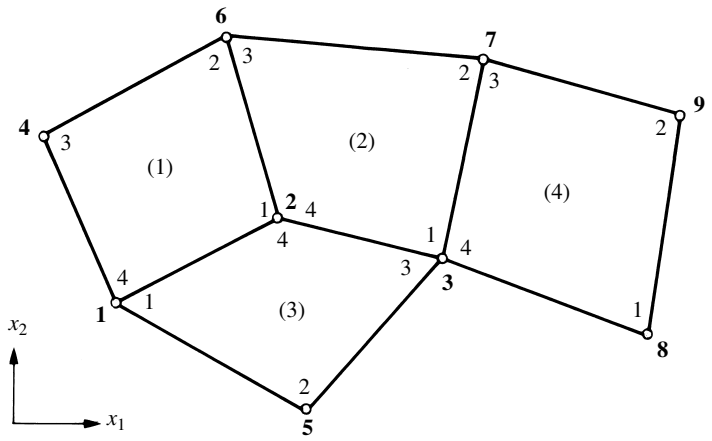
3.5.1 Grid and mesh notions

We call a ***grid*** the minimal description of the finite element discretization of the studied domain. It is composed of the coordinates of the nodes that make it up as well as the connectivity, that is, the manner in which the elements are formed. It is a purely geometric description.

In practice, the set of data related to the grid consists of the list of nodes and their coordinates as well as the element connection matrix which specifies each element by numbering the list of nodes of which it is composed. Figure 3.17 shows a simple example of a grid. Note that the list of nodes for each element is given counterclockwise indifferently to the first node in the list.

The ***mesh*** is a richer description than the grid of the approximation by finite elements. It contains supplementary information relative to the type of elements used, to integration, the boundaries, the boundary conditions, etc. Some of the information susceptible to being found in a mesh file are:

- The type of approximation chosen on each element (interpolation functions) as well as the type of integration (number of Gauss integration points, their position in the reference element and weights); this for the different fields approximated (interpolation is often performed differently for the velocity and pressure fields in fluid mechanics for example: cf (3.31) and (3.32)). In



Node coordinates

1	$x_1 = .$	$x_2 = .$
2		
3		
⋮	⋮	⋮
9		

connectivity

1:	2	6	4	1
2:	3	7	6	2
3:	1	5	3	2
4:	8	9	7	3

Figure 3.17 Example of a grid in 2 dimensions. The elements are linear quadrangles. The global number for each node is in bold type, the local number of the node in normal type, and the element number is in parentheses.

general, in the mesh file we use a code for each element type that refers to a library of elements which is subsequently read.

- The location of the boundary facets in 3 dimensions, or the edges in 2 dimensions. The number of the element containing the boundary facet (edge) i, as well as the local numbering of that facet (edge) in the element can also be found in the file.
- The symmetries. In 2 dimensions, the axes of symmetry are defined in the mesh file by the coordinates of a point on the axis and a direction vector, and the axis to which each boundary edge belongs can be specified. Similarly in 3 dimensions, where symmetry planes are defined by a point on the plane and the normal vector, the plane to which a boundary facet belongs can be specified.
- In the case where different types of elements are used simultaneously, the notion of a group of elements of the same type is employed. For each element, the mesh file then contains the number of the group to which it belongs.
- The same notion of groups of elements can also be used for computations mixing elements with different properties (computation with multiple materials). Similarly, it can be practical to group facets (or edges in 2 dimensions) to more easily impose boundary conditions on the domain.

The liberty of the programmer to define the data structure can be imagined. In particular, for the description of the boundaries and the boundary conditions, he will be tempted to introduce supplementary matrices into the mesh file that can help avoid repeated calculations later and thus to reduce the total computation time. This is one of the aspects of the definition of a ***data structure*** that affects the efficiency of a finite element computation. However, when two different computation programs are to be used on the same problem, the compatibility of the data structures needs to be enforced, which may require the development of interface programs for that purpose.

3.5.2 Assembly operations

Up to now we have seen that the spatial discretization of the weak formulation with finite elements, as for the finite difference method, results finally in the solution of a system of linear equations, possibly at the price of linearizing the initial problem if it is non linear (chap. 4). The matrix and the right hand side vector for this system are composed of integrals on the domain Ω or its boundary $\partial\Omega$ (cf (3.65) for example). In section 3.2 we have seen the procedure for the calculation of an integral on a finite element. The link between the local integral at the element level and the global integral is yet to be defined; this is the ***assembly operation***, which plays a central role in the implementation of the method. Its principle, which we will explain in detail below, is the following: the calculation of a global integral is subdivided into calculations of the contribution of each element, then the global problem is reconstituted to have the solution in terms of the nodal field.

Consider then a subdivision of the domain Ω into *Ne* elements each occupying a domain Ω^e. Each element possesses *nne* interpolation nodes *locally* numbered from 1 to *nne*. At each node numbered k locally, there is a corresponding global number ν_k^e. Thus, in the example in figure 3.17, we have for the element $e = 2$:

$$\nu_1^2 = 3 \qquad \nu_2^2 = 7 \qquad \nu_3^2 = 6 \qquad \nu_4^2 = 2 \tag{3.141}$$

Consider again the stationary diffusion example, but this time without advective transport, to be more clear. The discretized weak formulation results in a global system of equations, from (3.62):

$$c^j \left[\int_\Omega D\ \mathbf{grad}\,\psi_j \cdot \mathbf{grad}\,\psi_i \,\mathrm{d}V + \int_{\partial\Omega_N} \alpha \psi_i \psi_j \,\mathrm{d}S \right] = \int_{\partial\Omega_N} \alpha \bar{c}_a \psi_i \,\mathrm{d}S \tag{3.142}$$

for $i = 1, Nn$

We can write each integral as a sum of integrals calculated on the different elements. Then, if the volumetric and surface contributions of the system matrix $[\boldsymbol{K}]$

are separated, we have for the volumetric part, denoted $[\boldsymbol{K}^{\mathrm{vol}}]$:

$$K_{ij}^{\mathrm{vol}} = \int_{\Omega} D\ \mathbf{grad}\,\psi_j \cdot \mathbf{grad}\,\psi_i\ \mathrm{d}V = \sum_{e=1}^{Ne}\left(\int_{\Omega^e} D\ \mathbf{grad}\,\psi_j \cdot \mathbf{grad}\,\psi_i\ \mathrm{d}V\right) = \sum_{e=1}^{Ne} K_{ij}^{e,\,\mathrm{vol}} \tag{3.143}$$

It must be noted that in this expression, i and j are the *global* numbers for the nodes, and that the interpolation functions are extended at value zero into the elements that do not contain the nodes i and j. The term (i, j) in the integral over Ω^e is thus null if either of the two nodes i or j does not belong to the element e. Also, the term K_{ij}^{vol} is the sum of several element contributions if i and j are shared by several elements.

We then obtain the following strategy for the calculation of the matrix $[\boldsymbol{K}^{\mathrm{vol}}]$. On each element Ω^e, and for each pair of local node numbers (k, k') of element e, we calculate:

$$K_{kk'}^{e,\mathrm{vol}} = \int_{\Omega^e} D\ \mathbf{grad}\,\psi_{\nu_k^e} \cdot \mathbf{grad}\,\psi_{\nu_{k'}^e}\ \mathrm{d}V = \int_{\Omega^e} D\ \mathbf{grad}\,\psi_k^e \cdot \mathbf{grad}\,\psi_{k'}^e\ \mathrm{d}V \tag{3.144}$$

which is the term (k, k') of the elementary matrix $[\boldsymbol{K}^{e,\mathrm{vol}}]$. We then add this term with the already existing term $(\nu_k^e, \nu_{k'}^e)$ of the global matrix $[\boldsymbol{K}^{\mathrm{vol}}]$:

$$K_{\nu_k^e \nu_{k'}^e}^{\mathrm{vol}} + K_{kk'}^{e,\,\mathrm{vol}} \rightarrow K_{\nu_k^e \nu_{k'}^e}^{\mathrm{vol}} \tag{3.145}$$

This *assembly* operation allows the building of the global matrix $[\boldsymbol{K}^{\mathrm{vol}}]$ (of dimension $Nn \times Nn$) from the local matrices $[\boldsymbol{K}^{e,\,\mathrm{vol}}]$ (of dimension $nne \times nne$). The matrix $[\boldsymbol{K}^{\mathrm{vol}}]$ is then finally obtained after processing of all the elements $e = 1$, Ne.

The same strategy is obviously applied for the surface contribution and for the term on the right hand side. In the first case we have:

$$K_{ij}^{\mathrm{surf}} = \int_{\partial\Omega_N} \alpha\psi_i\psi_j\ \mathrm{d}S = \sum_{e=1}^{Ne}\left(\int_{\partial\Omega_N^e} \alpha\psi_i\psi_j\ \mathrm{d}S\right) = \sum_{e=1}^{Ne} K_{ij}^{e,\,\mathrm{surf}} \tag{3.146}$$

where $\partial\Omega_N^e$ designates the set of boundary faces of the element e belonging to the discretization of the surface $\partial\Omega_N$ under a Cauchy condition. If the element e has no boundary face its contribution to $[\boldsymbol{K}^{\mathrm{surf}}]$ is null.

Finally, for the right hand side term, one proceeds in the same way with:

$$b_i = \int_{\partial\Omega_N} \alpha\bar{c}_a \psi_i \, dS = \sum_{e=1}^{Ne} \left(\int_{\partial\Omega_N^e} \alpha\bar{c}_a \psi_i \, dS \right) = \sum_{e=1}^{Ne} b_i^e \tag{3.147}$$

$$b_{\nu_k^e} + \int_{\partial\Omega_N^e} \alpha\bar{c}_a \psi_k^e \, dS \rightarrow b_{\nu_k^e} \tag{3.148}$$

Practical Example

Consider the very simple mesh of 3.17, composed of four elements.

The assembly of a stiffness matrix [$\boldsymbol{K}$] (for a problem with scalar field such as chemical diffusion or heat transfer) is carried out according to the following steps:

a) Construction of the local matrix [$\boldsymbol{K}^1$], on element 1.
   [$\boldsymbol{K}^1$] is a 4×4 matrix for this linear quadrangular element.
   Local nodes: 1, 2, 3, 4. Global nodes: 2, 6, 4, 1.
b) Assembly of [$\boldsymbol{K}^1$] into [$\boldsymbol{K}$]. The global matrix [$\boldsymbol{K}$] has dimension 9×9 as there are 9 nodes in total.
   Examples of the operations performed: $K_{12}^1 \rightarrow K_{26}$; $K_{13}^1 \rightarrow K_{24}$, etc.
c) Local construction of [$\boldsymbol{K}^2$].
d) Assembly of [$\boldsymbol{K}^2$] into [$\boldsymbol{K}$].
e) Local construction of [$\boldsymbol{K}^3$].
f) Assembly of [$\boldsymbol{K}^3$] into [$\boldsymbol{K}$].
g) Local construction of [$\boldsymbol{K}^4$].
   Assembly of [$\boldsymbol{K}^4$] into [$\boldsymbol{K}$].

Figure 3.18 shows examples of the updating of terms in the matrix [$\boldsymbol{K}$] performed during the assembly for elements $e = 1$ and $e = 2$.

In general, this procedure is easily programmed following the schematic of figure 3.19.

Returning to the example of figure 3.17, the global matrix [$\boldsymbol{K}$] after assembly contains the following element contributions:

$$\left.\begin{aligned} K_{11} &= K_{44}^1 + K_{11}^3 \\ K_{12} &= K_{41}^1 + K_{14}^3 \\ K_{23} &= K_{41}^2 + K_{43}^3 \\ K_{22} &= K_{11}^1 + K_{44}^2 + K_{44}^3 \\ &\text{etc.} \end{aligned}\right\} \tag{3.149}$$

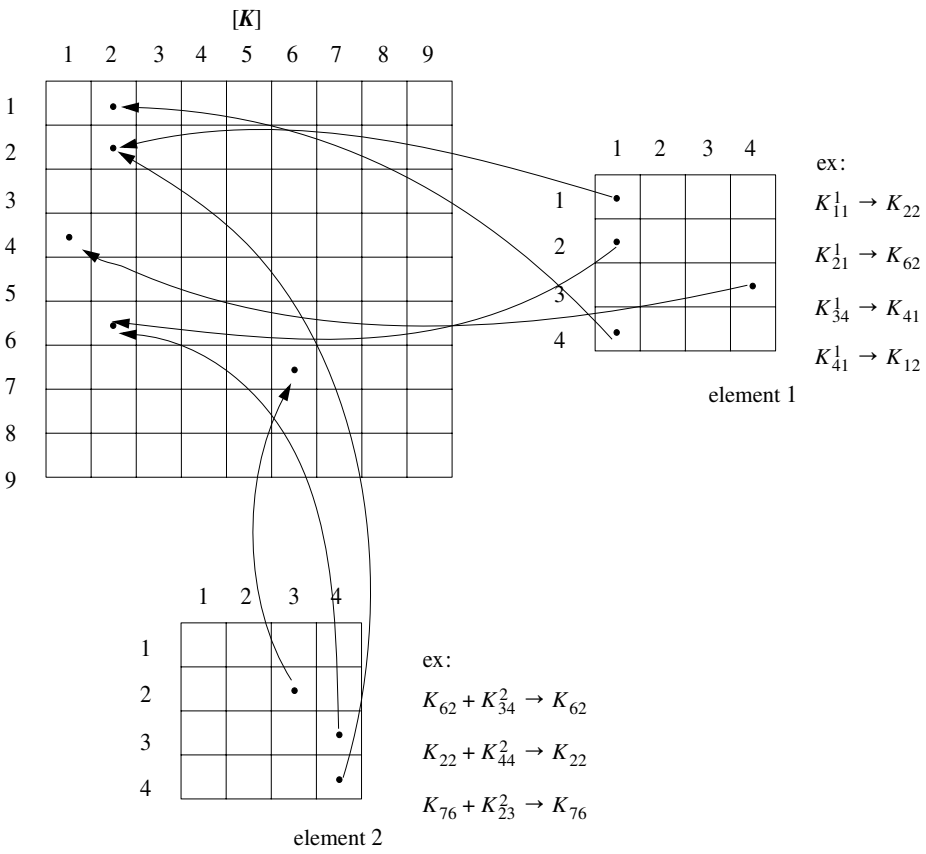


Figure 3.18 Illustration of the assembly of a stiffness matrix, in the case of an unknown *scalar* field, for the mesh of figure 3.17. Accumulation of the elementary matrix terms of the first two elements in the global matrix (only a few operations are indicated by the arrows).

Its general aspect is shown in figure 3.20.

Similarly, the assembly of the right hand side term gives, among other components:

$$\left.\begin{aligned} b_1 &= b_4^1 + b_1^3 \\ b_2 &= 0 \\ b_3 &= b_1^2 + b_3^3 + b_4^4 \\ b_4 &= b_3^1 \\ b_5 &= b_2^3 \\ &\text{etc.} \end{aligned}\right\} \qquad (3.150)$$

Initialisation $[\boldsymbol{K}] = 0$
Loop over the elements e

Transfer of the necessary information: global → local
Initialization of the elementary matrix $[\boldsymbol{K}^e] = 0$

Loop over the integration points p of the reference element

Calculate the determinant of the Jacobian (real element / reference element)
Loop over the local nodes k
Loop over local nodes k'
Calculate the term kk' at point p
Accumulate in the integrals $K^e_{kk'}$
End of loop over local nodes k'
End of loop over local nodes k

End of loop over integration points p

Assembly of $[\boldsymbol{K}^e]$ into $[\boldsymbol{K}]$

End of loop over elements e

Figure 3.19 Algorithm for calculation of the matrix in a finite element problem.

$[\boldsymbol{K}]$

1,3	1,3	3	1	3	1			
1,3	1,2,3	2,3	1	3	1,2	2		
3	2,3	2,3	–	3	2	2,4	4	4
1	1	–	1	–	1	–	–	–
3	3	3	–	2,3,4	–	–	–	–
1	1,2	2	1	–	1,2	2	–	–
	2	2,4	–	–	2	2,4	4	4
		4	–	–	–	4	4	4
		4	–	–	–	4	4	4

Figure 3.20 General aspect of the assembled stiffness matrix for the example presented in figure 3.17 (the numbers appearing in position (i, j) in the matrix are the numbers of the elements that contribute to the term K_{ij}).

3.5.3 Band structure of the matrices and the importance of the numbering

Referring to (3.142), note that all matrices in a finite element problem have a ***band structure*** around the diagonal, as shown in figure 3.21. This comes from the fact that nodes i and j must be neighbors – that is, belong to the same element – to give a non zero term to K_{ij}. As a consequence, on line i of the matrix, only zeros are found past the neighbor the furthest away from i in the sense of the numbering.

In addition, the band structure of the matrix is always symmetric: if a term K_{ij} is non zero (i.e. the nodes i and j both belong to at least one element), then the term K_{ji} is also non zero. Such matrices are called *symmetric profile* matrices. If in addition $K_{ij} = K_{ji}$ for every pair (i, j), the matrix is then symmetric.

The ***bandwidth*** of the matrix, denoted bw, for a problem with one unknown per node, is equal to $(bw_g + 1)$ where bw_g is the maximum difference between the number of two nodes in an element of the mesh. bw_g is called the *geometric bandwidth*. For example, in the case shown in figure 3.17, we have $bw_g = 6 = (9 - 3)$ for element 4. The bandwidth of the matrix is 7, as can be seen in figure 3.20.

We will see later on (chap. 4) that for a banded matrix, the time spent in factorization – which is necessary for the direct solution of a linear symmetric system of equations – is proportional to the product of the bandwidth squared times the number of unknowns. It is thus very important, for a given number of variables, to minimize the geometric bandwidth of a mesh in order to minimize the computation costs. For a given mesh topology, one must thus number the nodes in such a way that bw_g is minimized. Different algorithms have been developed for this purpose, for which we refer the reader to the bibliography at the end of the chapter. Thus, in the current case, the geometric bandwidth can be reduced from 6 to 4 with a better numbering scheme (fig. 3.22).

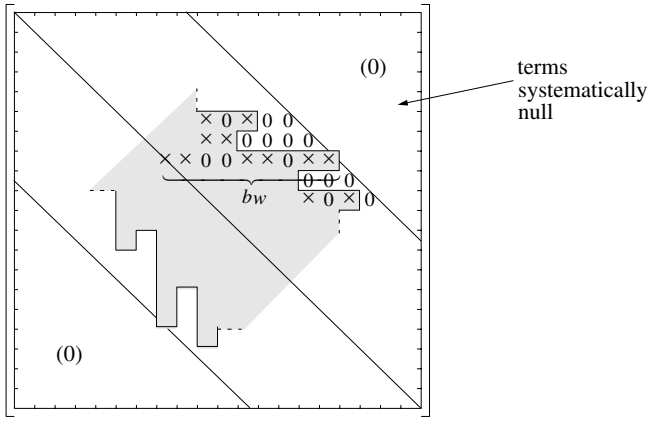
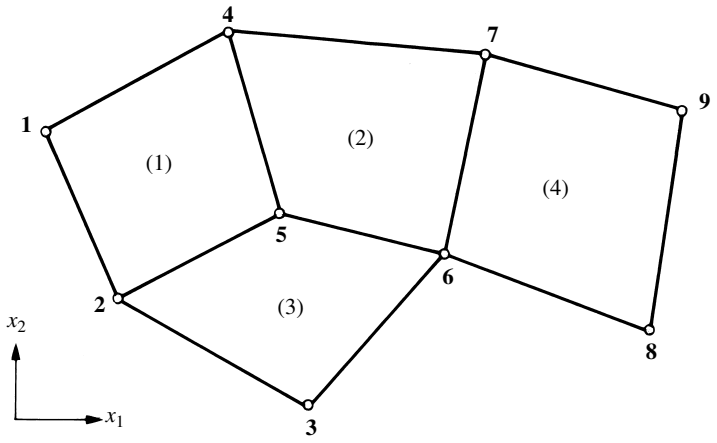


Figure 3.21 Band structure of finite element matrices (crosses symbolize non zero terms of the matrix).



×	×	0	×	×	0	0	0	0
×	×	×	×	×	×	0	0	0
0	×	×	0	×	×	0	0	0
×	×	0	×	×	×	×	0	0
×	×	×	×	×	×	×	0	0
0	×	×	×	×	×	×	×	×
0	0	0	×	×	×	×	×	×
0	0	0	0	0	×	×	×	×
0	0	0	0	0	×	×	×	×

Figure 3.22 Renumbering of the mesh nodes of figure 3.17 minimizing the bandwidth of the system matrix to be solved, and the general aspect of the matrix.

Note finally that in the case where many unknowns per node are considered (solution for a vector field, or simultaneous solution of many scalar or vector fields), one tends to think in terms of nodal degrees of freedom to define the band structure of the matrix and to calculate its bandwidth. Thus if N is the number of degrees of freedom per node, the bandwidth of the matrix for the problem, expressed in degrees of freedom, is given by: $bw = N\,(bw_g + 1)$.

3.5.4 Treatment of the boundary conditions

In the previous sections 3.3 and 3.4, the mechanism for the incorporation of non-essential boundary conditions (Neumann or Cauchy) in the integral form of the equations was explained. It was illustrated for a scalar advection-diffusion problem (a given normal gradient imposed, or a function of the unknown) and for a vector problem in mechanics (a given stress vector imposed, or a stress vector as a function of the unknown velocity field).

The case of an essential (Dirichlet) boundary condition which consists of directly imposing the value of some nodal degrees of freedom, remains to be treated.

In fact, the degrees of freedom imposed is generally not taken into account in the procedures for calculation and assembly of the stiffness matrix and the right hand side term of the system to solve. Thus in the scalar problem studied in section 3.3, the matrices and right hand side term are calculated for each element, and then assembled, without taking into consideration whether the degrees of freedom are imposed. Then, for all the degrees of freedom, and whatever their status, the components of the system matrix and the right hand side term are given by (3.65) and (3.67). Similarly, for the vector problem of section 3.4, the matrix and right hand side term are defined by (3.115) and (3.117), for all the degrees of freedom.

To introduce Dirichlet conditions, it is thus necessary to act directly on the matrix and the right hand side term of the system, or on the linearized stiffness matrix if the problem is non linear (sect. 4.5). We describe in the following the two most frequently used methods to treat such conditions.

Direct method

Suppose that we want to impose the value $\bar{c}^{\mathrm{i}}$ upon the degree of freedom i (here the variable denoted c can be either a scalar or a vector, and Nn is the number of degrees of freedom of the problem). The substitution of this value in the initial assembled system $[\boldsymbol{K}]\ \boldsymbol{c} = \boldsymbol{b}$ results in a new system of the same dimension as the first, where all the terms in which $\bar{c}^{\mathrm{i}}$ appear (therefore known) are moved to the right hand side:

$$\begin{bmatrix} K_{11} & \dots & 0 & \dots & K_{1,\mathrm{Nn}} \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ K_{\mathrm{Nn},1} & \dots & 0 & \dots & K_{\mathrm{Nn},\mathrm{Nn}} \end{bmatrix} \begin{bmatrix} c^{1} \\ \vdots \\ c^{\mathrm{i}} \\ \vdots \\ c^{\mathrm{Nn}} \end{bmatrix} = \begin{bmatrix} b_1 - K_{1\mathrm{i}}\bar{c}^{\mathrm{i}} \\ \vdots \\ \bar{c}^{\mathrm{i}} \\ \vdots \\ b_{\mathrm{Nn}} - K_{\mathrm{Nn},\mathrm{i}}\bar{c}^{\mathrm{i}} \end{bmatrix} \tag{3.151}$$

Other than the modifications to the right hand side, note those for the ith line and column of the matrix to obtain $c^{\mathrm{i}} = \bar{c}^{\mathrm{i}}$ for this degree of freedom after solution of the system. The procedure is continued this way for all the imposed values. This method is thus related to that introduced in chapter 2 for the FDM.

Penalty method

It is a bit simpler to implement because only the diagonal terms of $[\boldsymbol{K}]$ for which c is imposed are changed, and the modification of the right hand side term is easier. It consists of replacing the initial system by the following system:

$$\begin{bmatrix} K_{11} & \dots K_{1i} \dots & K_{1,Nn} \\ \vdots & \vdots & \vdots \\ K_{i1} & \dots K_{ii} + \chi_p \dots & K_{i,Nn} \\ \vdots & \vdots & \vdots \\ K_{Nn,1} & \dots K_{Nn,i} \dots & K_{Nn,Nn} \end{bmatrix} \begin{bmatrix} c^1 \\ \vdots \\ c^i \\ \vdots \\ c^{Nn} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i + \bar{c}^i \chi_p \\ \vdots \\ b_{Nn} \end{bmatrix} \tag{3.152}$$

where χ_p is "sufficiently" large with respect to the terms of the matrix $[\boldsymbol{K}]$. Take for example

$$\chi_p = 10^4 \operatorname*{Max}_{i,j=1,Nn} |K_{ij}| \tag{3.153}$$

The value $\bar{c}^i$ is obviously imposed on c^i with a higher and higher precision as χ_p becomes large. However, if an iterative solution of the system is employed (chap. 4), values of χ_p which are too large should be avoided as they could adversely affect the convergence rate.

In conclusion of this section on the implementation of the FEM, remember that the methods for the solution of systems of equations, linear or non linear, which are shared by the finite difference and the finite element methods, will be treated in chapter 4.

3.6 NON STATIONARY PROBLEMS

We focus in this section on detailing the solution with the finite element method of problems evolving in time. Such problems are obviously encountered very often in materials science. Consider, for example the diffusion of heat in material transformation processes, chemical diffusion at the root of segregation/homogenization phenomena in phase transformations, unsteady flows such as polymer or metal injection, etc.

This type of problem leads, as we are going to see with the example in 3.6.1, to a system of differential equations in time for which the solution requires a suitable choice of a temporal scheme, in a way similar to that seen for the finite difference method in chapter 2.

3.6.1 Example of non stationary diffusion

In the following, we again take the diffusion example already treated in chapter 2 and section 3.3. With respect to (3.52), a partial derivative with respect to time of the concentration of solute c appears when the problem is non stationary (1.74):

$$\frac{dc}{dt} = \frac{\partial c}{\partial t} + \boldsymbol{v} \cdot \mathbf{grad}\, c = \operatorname{div}(D\, \mathbf{grad}\, c) \tag{3.154}$$

As in the stationary case, natural boundary conditions (Neumann or Cauchy) given by (3.53) and (3.54), and essential (Dirichlet) conditions, are imposed respectively on the parts $\partial\Omega_N$ and $\partial\Omega_D$ of the domain under consideration Ω. Additionally, the temporal evolution of the concentration field requires the specification of the initial conditions $c(\boldsymbol{x}, t = 0)$ at every point in the domain (for example: uniform concentration $c = \bar{c}_0$). The domain Ω is discretized in finite elements with a mesh assumed to be fixed in space according to an Eulerian formulation (in other words, the flow "traverses" the mesh). Under these conditions, the temporal derivatives of c at the mesh nodes, which we denote $\dot{c}$ in this paragraph, are *partial* derivatives with respect to time.

As for the finite differences, the velocity field $\boldsymbol{v}$, assumed uniform, is a given of the problem. If we place ourselves in the context of simulating a process – for example the solidification of an ingot during which we would like to calculate the transport and diffusion of the alloy components – the velocity field $\boldsymbol{v}$ will no longer be constant on the domain under study, but may itself be interpolated at the mesh nodes and result from a separate mechanical computation of the flow of the metal (chap. 7).

Note finally that (3.154) is a generic equation for advection-diffusion phenomena. In particular, it can be applied to the heat transfer problem by replacing the concentration c by the temperature T and the diffusion coefficient D by the thermal diffusivity, $\kappa/\rho c_p$.

It is easy to show (section 3.3.2) that the application of a Galerkin weak formulation to (3.154) leads to the following system of equations:

$$\int_\Omega \psi_i \frac{\partial c}{\partial t}\, \mathrm{d}V + \int_\Omega \psi_i \boldsymbol{v} \cdot \mathbf{grad}\, c\, \mathrm{d}V + \int_\Omega D\, \mathbf{grad}\, c \cdot \mathbf{grad}\, \psi_i\, \mathrm{d}V = -\int_{\partial\Omega_N} \alpha \psi_i (c - \bar{c}_a)\, \mathrm{d}S \qquad \text{for } i = 1, Nn \tag{3.155}$$

which again can be written, introducing completely nodal interpolation (3.60):

$$c^j \left[\int_\Omega D\, \mathbf{grad}\, \psi_j \cdot \mathbf{grad}\, \psi_i\, \mathrm{d}V + \int_{\partial\Omega_N} \alpha \psi_i \psi_j\, \mathrm{d}S + \int_\Omega \psi_i \boldsymbol{v} \cdot \mathbf{grad}\, \psi_j\, \mathrm{d}V \right] + \dot{c}^j \left[\int_\Omega \psi_i \psi_j\, \mathrm{d}V \right] = \int_{\partial\Omega_N} \alpha \bar{c}_a \psi_i\, \mathrm{d}S \qquad \text{for } i = 1, Nn \tag{3.156}$$

We thus arrive at the following system of differential equations:

$$[\boldsymbol{M}]\dot{\boldsymbol{c}} + [\boldsymbol{K}]\boldsymbol{c} = \boldsymbol{b} \tag{3.157}$$

in which $\boldsymbol{c}$ and $\dot{\boldsymbol{c}}$ are the column vectors having the nodal concentrations and their temporal derivatives, respectively, for components. The stiffness matrix [$\boldsymbol{K}$] and the right hand side vector $\boldsymbol{b}$ do not change with respect to the stationary formulation (cf (3.65) and (3.67)). The matrix [$\boldsymbol{M}$], called the ***mass matrix***, has the following components:

$$M_{ij} = \int_{\Omega} \psi_i \psi_j \, dV \tag{3.158}$$

This symmetric matrix has the same profile as the matrix [$\boldsymbol{K}$] and is assembled in an identical way.

Condensation of the mass matrix

For each element e the elementary mass matrix has for components in two dimensions:

$$M^e_{ij} = \int_{\Omega^e} \psi^e_i \psi^e_j \, dV = \int_{\Omega^r} \psi^r_i \psi^r_j \, \det[\boldsymbol{J}^e(\boldsymbol{\xi})] \, d\xi_1 \, d\xi_2 \tag{3.159}$$

For the particular case of a linear triangular element, the Jacobian matrix [$\boldsymbol{J}^e$] is constant (section 3.2.8) and we have:

$$M^e_{ij} = \det[\boldsymbol{J}^e] \int_0^1 d\xi_1 \int_0^{1-\xi_1} \psi^r_i \psi^r_j \, d\xi_2 \tag{3.160}$$

which by exact integration of the products of the interpolation functions gives (exercise 3.7.4):

$$[\boldsymbol{M}^e] = \frac{\det[\boldsymbol{J}^e]}{2} \begin{bmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \end{bmatrix} \tag{3.161}$$

The term appearing in front of the matrix is none other than the surface (the "mass") of the triangle as,

$$\det[\boldsymbol{J}^e] = (x_1^2 - x_1^1)(x_2^3 - x_2^1) - (x_2^2 - x_2^1)(x_1^3 - x_1^1) \tag{3.162}$$

In the ***lumped mass matrix*** approximation, the extra-diagonal terms are summed on the diagonal, which gives then:

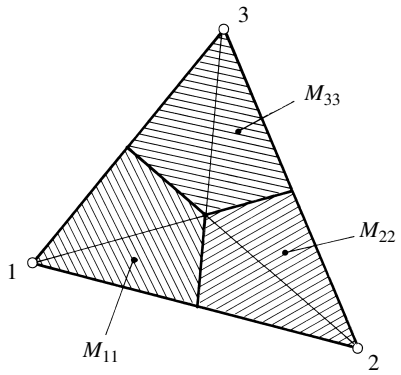


Figure 3.23 Assignment of a third of the surface of a linear triangle to the three vertex nodes, following the partition given by the medians, and corresponding to the condensation process of the mass matrix in two dimensions.

$$[\boldsymbol{M}^{\mathrm{e}}] = \frac{\det[\boldsymbol{J}^{\mathrm{e}}]}{2}\begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \tag{3.163}$$

This amounts to attributing a third of the surface (mass) of the triangle to each node (fig. 3.23). It should be noted that the condensation operates in the same way for linear or quadratic quadrangular elements, but that it is not so simple for quadratic triangles, where a modification of the interpolation functions is necessary to avoid a singular condensed matrix.

The mass matrix thus being calculated, and maybe condensed, it is now a question of solving the differential system (3.157) in order to calculate the evolution of the concentration at the mesh nodes. There are many temporal integration schemes to solve this type of system. We limit our discussion to single step "*θ-method*" types, as already introduced in the context of finite differences (chap. 2).

3.6.2 Single step "θ-*method*" *type scheme*

In a manner analog to that which was done in chapter 2, assume that the solution is known up to time t. We want to calculate the new nodal concentrations $\boldsymbol{c}^{\mathrm{t}+\Delta\mathrm{t}}$ at time $t + \Delta t$. This type of scheme consists of performing a temporal discretization of (3.157), of the following form:

$$[\boldsymbol{M}]\frac{1}{\Delta t}(\boldsymbol{c}^{\mathrm{t}+\Delta\mathrm{t}} - \boldsymbol{c}^{\mathrm{t}}) + [\boldsymbol{K}](\theta\boldsymbol{c}^{\mathrm{t}+\Delta\mathrm{t}} + (1-\theta)\boldsymbol{c}^{\mathrm{t}}) = \theta\boldsymbol{b}^{\mathrm{t}+\Delta\mathrm{t}} + (1-\theta)\boldsymbol{b}^{\mathrm{t}} \tag{3.164}$$

with

$\boldsymbol{c}^{t}$	vector of concentrations at the mesh nodes at time t,
$\boldsymbol{c}^{t+\Delta t}$	vector of concentrations at the mesh nodes at time $t + \Delta t$,
$\boldsymbol{b}^{t}$	right hand side vector at t,
$\boldsymbol{b}^{t+\Delta t}$	right hand side vector at $t + \Delta t$.
θ	parameter with value between 0 and 1.

Before starting the solution of the system (3.164) for $\boldsymbol{c}^{t+\Delta t}$, we detail three of the main schemes used in practice:

- $\theta = 0$ corresponds to the ***Euler explicit scheme***, which can be decomposed into two stages. In the first stage, the temporal derivative of $\boldsymbol{c}$ at time t is expressed, by writing (3.157) at time t:

$$\dot{\boldsymbol{c}}^{t} = [\boldsymbol{M}]^{-1}(-[\boldsymbol{K}]\boldsymbol{c}^{t} + \boldsymbol{b}^{t}) \tag{3.165}$$

In the second stage, we integrate on the time interval with this derivative, considering it constant, to obtain $\boldsymbol{c}^{t+\Delta t}$:

$$\boldsymbol{c}^{t+\Delta t} = \boldsymbol{c}^{t} + \Delta t\dot{\boldsymbol{c}}^{t} \tag{3.166}$$

The combination of the two expressions above results in the following system,

$$\frac{1}{\Delta t}[\boldsymbol{M}]\boldsymbol{c}^{t+\Delta t} = \left(\frac{1}{\Delta t}[\boldsymbol{M}] - [\boldsymbol{K}]\right)\boldsymbol{c}^{t} + \boldsymbol{b}^{t} \tag{3.167}$$

which is seen to be none other than (3.164) with $\theta = 0$. The explicit scheme is called the forward marching scheme, because the derivative for the time increment is calculated from the concentrations $\boldsymbol{c}^{t}$ known at time t.

- $\theta = 1$ corresponds to the ***Euler implicit scheme***, which consists this time of integrating the derivative expressed at the end of the increment over the time interval (the backward marching scheme), thus introducing the unknown $\boldsymbol{c}^{t+\Delta t}$:

$$\dot{\boldsymbol{c}}^{t+\Delta t} = [\boldsymbol{M}]^{-1}(-[\boldsymbol{K}]\boldsymbol{c}^{t+\Delta t} + \boldsymbol{b}^{t+\Delta t}) \tag{3.168}$$

$$\boldsymbol{c}^{t+\Delta t} = \boldsymbol{c}^{t} + \Delta t\dot{\boldsymbol{c}}^{t+\Delta t} \tag{3.169}$$

The system to solve is thus:

$$\left(\frac{1}{\Delta t}[\boldsymbol{M}] + [\boldsymbol{K}]\right)\boldsymbol{c}^{t+\Delta t} = \left(\frac{1}{\Delta t}[\boldsymbol{M}]\right)\boldsymbol{c}^{t} + \boldsymbol{b}^{t+\Delta t} \tag{3.170}$$

- $\theta = 1/2$ corresponds to the ***Crank-Nicolson scheme***, for which the temporal derivative of $\boldsymbol{c}$ over the time interval is taken equal to the arithmetic mean of the

derivatives at times t and $t + \Delta t$. Then, by combination of (3.165), (3.168) and the following relation,

$$\boldsymbol{c}^{t+\Delta t} = \boldsymbol{c}^{t} + \Delta t \frac{1}{2}(\dot{\boldsymbol{c}}^{t} + \dot{\boldsymbol{c}}^{t+\Delta t}) \tag{3.171}$$

we obtain the system to solve:

$$\left(\frac{1}{\Delta t}[\boldsymbol{M}] + \tfrac{1}{2}[\boldsymbol{K}]\right)\boldsymbol{c}^{t+\Delta t} = \left(\frac{1}{\Delta t}[\boldsymbol{M}] - \tfrac{1}{2}[\boldsymbol{K}]\right)\boldsymbol{c}^{t} + \tfrac{1}{2}(\boldsymbol{b}^{t+\Delta t} + \boldsymbol{b}^{t}) \tag{3.172}$$

While the explicit and implicit Euler schemes have a local error which goes as the time step to the second power, this scheme has an error which goes as the third power. Globally, the Euler schemes are to first power in the time step, while the Crank-Nicolson scheme goes as the square, and is thus more accurate.

In general, for $\theta \in [0,1]$, the system to solve for $\boldsymbol{c}^{t+\Delta t}$ is then the following:

$$\left(\frac{1}{\Delta t}[\boldsymbol{M}] + \theta[\boldsymbol{K}]\right)\boldsymbol{c}^{t+\Delta t} = \left(\frac{1}{\Delta t}[\boldsymbol{M}] - (1-\theta)[\boldsymbol{K}]\right)\boldsymbol{c}^{t} + \theta\boldsymbol{b}^{t+\Delta t} + (1-\theta)\boldsymbol{b}^{t} \tag{3.173}$$

This linear system is to be solved by a direct or iterative method (chap. 4). It should be noted that the solution can become trivial if an explicit scheme ($\theta = 0$) and a condensed mass matrix $[\boldsymbol{M}]$ are used. The system matrix is then diagonal and the Nn equations become independent. The solution is obviously very economical in computation time. However, as we will see in paragraph 3.6.3, the scheme is conditionally stable.

3.6.3 Study of the stability of the scheme

The stability of the scheme comes from the properties of the homogeneous system (where $\boldsymbol{b} = \boldsymbol{0}$), which we write:

$$\boldsymbol{c}^{t+\Delta t} = [\boldsymbol{G}]\boldsymbol{c}^{t} \tag{3.174}$$

For the scheme to be stable, that is, for $\boldsymbol{c}^{t+m\Delta t}$ to be bounded as m goes to infinity, it is necessary that the maximal absolute value of the eigenvalues λ_i of the matrix $[\boldsymbol{G}]$ of the homogeneous system be less than 1. This maximum value, called the ***spectral radius***, is denoted $\rho([\boldsymbol{G}])$:

$$\rho([\boldsymbol{G}]) = \underset{i=1,\,Nn}{\mathrm{Max}} |\lambda_i| \tag{3.175}$$

In the case under consideration here, the expression of the matrix $[\boldsymbol{G}]$ is easily obtained by multiplying the two sides of the system (3.173) by $\Delta t[\boldsymbol{M}]^{-1}$:

$$[\boldsymbol{G}] = ([\boldsymbol{I}] + \Delta t\theta[\boldsymbol{M}]^{-1}[\boldsymbol{K}])^{-1}([\boldsymbol{I}] - \Delta t(1-\theta)[\boldsymbol{M}]^{-1}[\boldsymbol{K}]) \tag{3.176}$$

The eigenvalues of $[\boldsymbol{G}]$, λ_i, are thus deduced from the eigenvalues Λ_i of the matrix $[\boldsymbol{M}]^{-1}[\boldsymbol{K}]$ by the relationships:

$$\lambda_i = \frac{1-(1-\theta)\Delta t\Lambda_i}{1+\theta\Delta t\Lambda_i} \tag{3.177}$$

The eigenvalues Λ_i are positive by the forms of $[\boldsymbol{M}]$ and $[\boldsymbol{K}]$. Therefore, the reader can easily verify that the condition on the spectral radius of the matrix $[\boldsymbol{G}]$ ($\rho([\boldsymbol{G}] < 1$) is equivalent to:

$$1 - 2\theta < \frac{2}{\Delta t\Lambda_i} \text{ for } i = 1, Nn \tag{3.178}$$

The stability of the scheme depends then on the value chosen for θ. For $\theta < 1/2$, it is seen that the preceding equation imposes a condition on the time step; the maximum value of Δt being given by:

$$\Delta t_{max} = \frac{2}{(1-2\theta)\rho([\boldsymbol{M}]^{-1}[\boldsymbol{K}])} = \frac{2}{(1-2\theta)\underset{i=1,Nn}{\text{Max}}\,\Lambda_i} \tag{3.179}$$

It is said then that the scheme is ***conditionally stable***. On the other hand, the condition (3.178) is always valid for $\theta \geq 1/2$ as $1 - 2\theta$ is then negative. In this case, the scheme is said to be ***unconditionally stable*** (no condition on the time step). In practice, one uses preferentially the schemes with a marked implicit character ($\theta \geq 1/2$).

Of course, the results above can be generalized to problems with many degrees of freedom per node by replacing Nn with the global number of degrees of freedom of the mesh (NnN).

3.6.4 Non-linearity and non stationarity

The origins of the non linearities in the chemical or thermal diffusion problems are multiple. First of all there is the dependence of the diffusivity on the solute concentrations (or temperature, respectively). As for the thermal transfer, one adds radiation type boundary conditions as well as phase transformations.

The implementation of a semi-implicit scheme with parameter θ is expressed by the following two equations:

$$\begin{aligned}\dot{\boldsymbol{c}}^{t+\theta\Delta t} = \theta\dot{\boldsymbol{c}}^{t+\Delta t} + (1-\theta)\dot{\boldsymbol{c}}^{t} &= \theta([\boldsymbol{M}]^{t+\Delta t})^{-1}(-[\boldsymbol{K}]^{t+\Delta t}\boldsymbol{c}^{t+\Delta t} + \boldsymbol{b}^{t+\Delta t}) \\ &+ (1-\theta)([\boldsymbol{M}]^{t})^{-1}(-[\boldsymbol{K}]^{t}\boldsymbol{c}^{t} + \boldsymbol{b}^{t})\end{aligned} \tag{3.180}$$

and

$$\boldsymbol{c}^{t+\Delta t} = \boldsymbol{c}^{t} + \Delta t\dot{\boldsymbol{c}}^{t+\theta\Delta t} \tag{3.181}$$

in which the mass matrix $[\boldsymbol{M}]$ is constant if the mesh does not evolve. As for the matrix $[\boldsymbol{K}]^{t+\Delta t}$ and the right hand side term $\boldsymbol{b}^{t+\Delta t}$, they depend on the unknown $\boldsymbol{c}^{t+\Delta t}$.

Thus the solution $\boldsymbol{c}^{t+\Delta t}$ can only be *directly* calculated in the purely explicit case ($\theta = 0$), which has the stability conditions seen above.

For the case of a partially implicit scheme ($\theta > 0$), there is no direct solution. For example, one is led, for a constant mass matrix, to zero the following residual vector, a function of $\boldsymbol{c}^{t+\Delta t}$:

$$\begin{aligned}\boldsymbol{R}(\boldsymbol{c}^{t+\Delta t}) \;=\; [\boldsymbol{M}](\boldsymbol{c}^{t+\Delta t} - \boldsymbol{c}^{t}) + \Delta t(\theta \boldsymbol{b}^{t+\Delta t} + (1-\theta)\boldsymbol{b}^{t} - \theta[\boldsymbol{K}]^{t+\Delta t}\boldsymbol{c}^{t+\Delta t} \\ -(1-\theta)[\boldsymbol{K}]^{t}\boldsymbol{c}^{t})\end{aligned} \tag{3.182}$$

The iterative methods to solve this type of system of non linear equations are detailed in chapter 4.

3.6.5 Introduction to the specific treatment of the advective transport terms

Note at this stage that if we envision a Lagrangian formulation to solve (3.154), that is, if the mesh nodes move at the fluid velocity, there is then equality between the partial derivative at the nodes of the mesh and the total derivative. The (non-symmetric) term in $\boldsymbol{v} \cdot \mathbf{grad}\ \psi$ disappears in (3.156) and $\dot{\boldsymbol{c}}$ is thus the vector of total derivatives (or material derivatives) at the nodes. This type of formulation is frequently utilized to model the heat transfer in deformation mechanics, for example to simulate the non stationary processes of forming, stamping and forging of metals, thermoforming and blowing of polymers.

Conversely, the modeling of non stationary fluid flows or stationary processes of forming (rolling of metals, drawing of polymer films …) is most often approached with an Eulerian formulation with a computation domain Ω, and thus a finite element discretization fixed in space. However, in this case, the Galerkin formulation proves to be poorly adapted when the convection is much larger than the diffusion (high Péclet numbers, chap. 2), leading to numerical oscillations in the solution. Beyond the simple reduction of the discretization spacing, which can lead to prohibitive computation time in practice, we will present the specific means to treat this problem in chapter 7.

3.7 EXERCISES

3.7.1 Prove the result (3.51) for the integration of the first derivative of a function on a linear triangle. Treat the case of a quadratic triangle with six nodes.

3.7.2 Volume calculation. We consider the volume Ω bounded by the closed surface S defined by a set of surface finite elements (triangles or quadrangles in space,

for example). Using the divergence theorem, give an example of a very simple vector function for which the integral of the flux across S is equal to the measure of Ω. Discuss the case where S is not a closed surface.

3.7.3 Apply the developments made in this chapter for the chemical diffusion case to heat transfer.

3.7.4 Find the equation (3.161) for the mass matrix for a linear triangular finite element.

3.7.5 In a chemical diffusion problem solved by finite elements, the contribution of the diffusion to the stiffness matrix of an element e is given by (3.65), that is, merging the local and global node numbers for simplicity:

$$K_{ij}^{e} = \int_{\Omega^e} D\,\mathbf{grad}\,\psi_i \cdot \mathbf{grad}\,\psi_j \,\mathrm{d}V = \int_{\Omega^e} D\frac{\partial \psi_i}{\partial x_k}\cdot\frac{\partial \psi_j}{\partial x_k}\,\mathrm{d}V$$

where D is the coefficient of chemical diffusion. For a linear triangular element, we denote:

$$\beta^i = y^j - y^k \qquad \text{and} \qquad \gamma^i = x^k - x^j$$

(i, j, k) indicating a circular permutation of the three nodes at the vertices (1,2,3). Expressing the Jacobian matrix of the element as $[\boldsymbol{J}^e]$, show that K_{ij}^{e} can also be written:

$$K_{ij}^{e} = \frac{D}{2\det[\boldsymbol{J}^e]}(\beta^i\beta^j + \gamma^i\gamma^j)$$

3.8 BIBLIOGRAPHY

K. J. Bathe, *Finite Element Procedures*, Prentice Hall, Englewood Cliffs, New Jersey, 1996

P. Ciarlet, *The Finite Element Method for Elliptic Problems*, North Holland, New York, 1978

T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, McGraw-Hill, New York, 2000

R.W. Lewis, K. Morgan, H.R. Thomas, K.N. Seetharama, *The Finite Elemente Method in Heat Transfer Analysis*, J. Wiley, New York, 1996

J.N. Reddy, *An Introduction to the Finite Element Method*, Mc Graw-Hill, New York, 1985

F. Thomasset, *Implementation of Finite Element Methods for Navier-Stokes Equations*, Springer, New York, 1981

R.E. White, *An Introduction to the Finite Element Method with Application to Non-Linear Problems*, John Wiley and Sons, New York, 1985

O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method*, Mc Graw-Hill, New York, 1989